ANALYZING SYSTEMS INTEGRATION BEST PRACTICES AND ASSESSMENT IN DOD SPACE SYSTEMS ACQUISITION

THESIS

Douglas R. Dillon, DAFC
Enriqueta Mariela Styers, DAFC

AFIT/GSE/ENV/09-02DL

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GSE/ENV/09-02DL

ANALYZING SYSTEMS INTEGRATION BEST PRACTICES AND ASSESSMENT IN DOD SPACE SYSTEMS ACQUISITION

THESIS

Presented to the Faculty

Department of Systems and Engineering Management

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Systems Engineering

Douglas R. Dillon, MS   Enriqueta Mariela Styers, BS

December 2009

AFIT/GSE/ENV/09-02DL

ANALYZING SYSTEMS INTEGRATION BEST PRACTICES AND ASSESSMENT IN DOD SPACE SYSTEMS ACQUISITION

Douglas R. Dillon, MS

Enriqueta Mariela Styers, BS

____________________________________ ____________

Dr. John M. Colombi (Chairman) Date

____________________________________ ____________

Dr. David R. Jacques (Member) Date

____________________________________ ____________

Dr. Joseph R. Wirthlin, LtCol, USAF (Member) Date

AFIT/GSE/ENV/09-02DL

Abstract

Senior leadership of the Air Force's Space and Missile Center suggested an investigation of systems integration within the space acquisition community in the fall of 2008. This thesis performs that investigation. A review concluded that while Systems Integration (SI) is extensively discussed as an area deserving considerable attention in the Systems Engineering literature, definitions are weak and methods and tools non-existent. Known SI activities are not being traced and assessed for adequacy throughout system development. Employing the Space System Acquisition Lifecycle Framework as the environment for this research, a method of characterizing and tracing SI throughout a program's lifecycle by using technical reviews and audits (TR&A) is proposed. Subsequent to a SI trace of an acquisition program, an assessment can be performed to determine the adequacy of the integration of Systems Engineering (SE) tasks. Using this assessment, prudent adjustments to program resources (e.g., SE, finance, research and development, program management, etc.) can be considered that will mitigate or resolve program deficiencies caused by insufficient SI. The proposed method is demonstrated across technical reviews and audits of the Global Positioning Systems (GPS) program. The results of this thesis should accentuate the value of SI during space system acquisition – a key consideration which is rarely recognized.

Acknowledgments

We would like to express our sincere appreciation to our faculty advisor, Dr. John Colombi, for his guidance and support throughout the course of this thesis effort. The insight and experience was certainly appreciated. We would, also, like to thank our sponsors Mr. Louis M. Johnson and Mr. James C. Breidenbach; all from the Space & Missiles Systems Center (SMC) for their support and latitude provided to us in this endeavor. Finally, we gratefully acknowledge colleagues who spent time listening and constructively criticizing our work which enhanced the clarity of our thesis.

Douglas R. Dillon

Enriqueta Mariela Styers

Table of Contents

Page

Page

List of Figures

Page

List of Tables

Page

ANALYZING SYSTEMS INTEGRATION BEST PRACTICES IN DOD SPACE SYSTEM ACQUISITION

I. Introduction

Background

Since the Industrial Revolution (circa 1760 - 1850), the size and complexity of systems has grown at an exponentially increasing rate. At the beginning of this era, a single person could comprehend and master the complexities of a system (e.g., the steam engine, the cotton gin, etc.). In today's world, however, it is exceedingly rare to find a person that could devote enough time and energy to comprehend and master a single, moderately complex, system. It now takes the time, energy and expertise of a small army of highly educated, trained and experienced professionals to successfully design or develop a modern system.

Over the past several decades, the engineering community has developed and fostered the field of Systems Engineering (SE). During this time, the SE discipline has evolved to address the entire technical effort required to develop and validate an integrated and total life cycle balanced system of people, processes, and products that satisfy modern, technologically advanced, systems. However, the individual engineering disciplines required to adequately support a Department of Defense (DoD) Space System Acquisition have rarely been appropriately integrated; resulting in continuing technical and programmatic shortfalls.

Numerous challenges continue to afflict DoD Space System Acquisition programs. These challenges are commonly attributed to the increasing technology capabilities available to choose from, user requirements to satisfy, suppliers to select, enterprise processes to integrate, and specific system security required for DoD Space System Acquisition programs. Any combination of these variable complexities can cause inadequate risk management, poor estimation planning, deficient SE continuity, lack of teamwork, poor communications and coordination, and insufficient monitoring of Systems Engineering & Integration (SE&I) progress. Resolving these program and technical maladies first requires an accurate assessment of the SI being conducted in a program (i.e., DoD Space System Acquisition program). Tracing Systems Integration (SI) in a DoD Space System Acquisition program, the subject of this thesis is the first step in curing these maladies.

Scanning through Government Accountability Office (GAO) testimonies and reports for Space System Acquisition programs also revealed a variety of problem areas related to SI. Additionally, there are SI proceedings from the repertoire of the International Council on Systems Engineering (INCOSE) that provide evidence that SI is a serious topic of discussion. SI remains a topic of interest within the SE community despite the lack of standards that enable the uniform selection of mechanisms to cope with these SI challenges.

Needless to say, these SI drivers set-up a landscape for "integrating the work of many people in different functional disciplines, working on different product system components, in many different process steps over time" (21:42). This landscape defines the three fundamental SI elements as *people, process, and product*. When working together, these SI elements decompose into integration areas where *interrelations,* *interactions, and interfaces* among and between these elements can be measured. Figure 1 illustrates these relationships using a conceptual model for integrating SI elements that yield a total of seven locations of integration activity – the patterned areas illustrated in this figure where SI occurs.

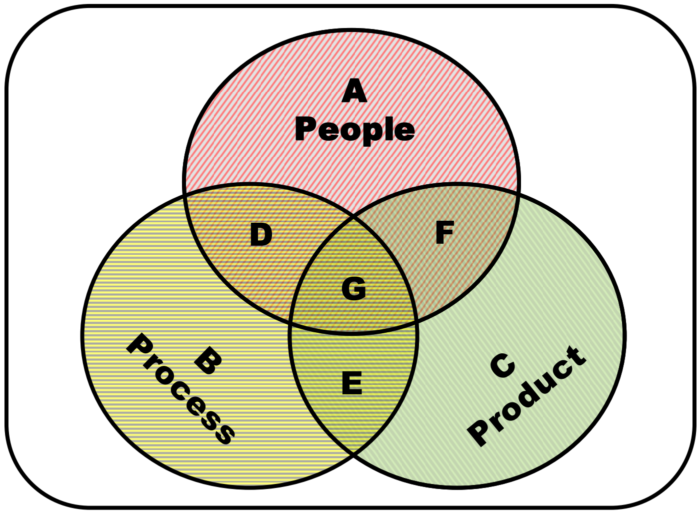

Figure 1. Basic Systems Integration Model

Purpose & Scope

The focus of this thesis is the development of a methodology in which program SI can be gauged from standard DoD Systems Acquisition Technical Reviews and Audits (TR&A) conducted for a DoD Space System Acquisition program. Case studies performed on these types of programs will be used to develop a methodology; which can be used to trace the application of SI within a program. Practitioners of this method will be able to assess the presence and the timing of SI being performed in a given program.

Research Objectives/Investigative Questions

To better understand the role and impact of SI in the DoD Space System

Acquisition Lifecycle Framework, this research pays particular attention to lessons learned that reveal characteristics related to integrating (i.e. *interrelating, interacting, or interfacing*) the system elements of *people, processes, and/or products*; all or any combination of which may have to work together. Capturing the program attributes that will be used to assess SI activity occurring in a Space System Acquisition program will be accomplished using the traditional series of TR&A which are conducted at logical transition points throughout the lifecycle framework of real-world programs; using case studies to obtain program artifacts. The following investigative questions provide a means to achieving the main objective:

(1) What are the characteristics of SI as revealed by lessons learned from various space-system programs?

(2) How are SI characteristics (attributes) traceable in the standard TR&A used in the Defense Acquisition System (37:34-42)?

Overview of Methodology

In pursuit of answering the investigative questions, research is conducted in two parts: the identification and categorization of SI characteristics, and the application of these SI characteristics to a DoD Space System Acquisition program in a manner that will support program SI traceability. Details of these methodology parts are described in Chapter III of this thesis.

In the first research area, a content analysis (10:1) is performed on 100 space-system related root-cause problem reports collected over the years of 1985-2005 (13:1-100). In each report, there is an average of four lesson-learned statements. For each lesson-learned statement, the presence of key concepts that have a predictable correlation to the SI model elements of *people, processes and products* are determined. Then, the linking relationships (i.e. *interrelating, interacting, or interfacing*) of these concepts are categorized into the seven integration areas of people-people (A), process-process (B), product-product (C), people-process (D), process-product (E), product-people (F), and/or people-process-product (G). This part of the thesis research was a necessary first step to better understand the elemental characteristics of SI, and to establish criteria that can be used to identify and trace the application of SI throughout a DoD Space System Acquisition program's life cycle.

During the second part of this research, a traceability analysis is performed on a set of TR&A that established linking relationships between/among the seven integration areas. Data collected during this part of the research focused on the SI concept-elements output from the previous part of the research; known as "high frequency tallied" SI concept-elements derived from the content analysis of the space systems lessons learned performed in the first part of this research. A traceability matrix model identifying the lifecycle phase(s) inferred by the high frequency tallied SI concept-elements was constructed next. This traceability matrix provides a means to document the tracing of the proposed SI model elements and their SI areas. In Chapter IV the traceability matrix model is then deployed using real-world space-system programs (i.e., case studies) that have completed, or are going through the TR&A process. Results of the TR&A matrix will then be used to validate the efficacy of using program TR&A for tracing SI characteristics.

Assumptions/Limitations

The identification of SI characteristics is subject to the authors' interpretation of the lesson-learned statements. This interpretive process assumes key concepts are accurately correlated to the SI areas used for the model element determination. Although lesson-learned statements may have been written by the space system analyst with subconscious biases and predispositions, there is no way to discern if biases have influenced the lesson-learned descriptions. Therefore, the assumption is that the authors of this thesis have accurately correlated key concepts with SI model elements. The primary limitation that this thesis encountered is the access to and availability of program or system information. Much of this information was considered by the Program Office to be sensitive or proprietary and not authorized for external review. As a secondary limitation, the tailoring of the technical review objectives, review criteria and/or supporting documentation degraded the SI analytical usefulness of these artifacts.

Preview of Thesis Composition

It has become a recognized fact within the Air Force acquisition community that the practice of life cycle Systems Engineering suffers from numerous shortcomings.

> *"Increasingly, I'm convinced that the systemic problem is in the field of systems engineering."* (41:1)

> *"An immediate transformation imperative for all programs is to focus more attention on the application of Systems Engineering principles and practices throughout the system life cycle."* (44:1)

Over time, within the DoD Space Systems acquisition community, the individual SE disciplines have matured and are highly reliable and effective. The holistic and seamless integration of these SE disciplines, however, have proven to be a substantially more challenging endeavor.

> *"The SBIRS case provides impetus to assess the level and quality of the integration of systems engineering and software systems engineering in ongoing programs."* (15:33)

The ability of a Program Manager or Chief Engineer to trace SI through the life-cycle phases of a DoD Space Systems Acquisition will provide valuable insight into the integration of the multitude of SE activities occurring throughout a space-system acquisition life cycle.

This thesis is primarily intended for use by the Space System Acquisition community to improve processes related to acquiring effective, affordable and timely space-systems. Its main audience is expected to be systems engineers and program managers who wish to expand their knowledge so as to deal with SE integration challenges. The chapters are organized as:

- Chapter I sets the stage and provides an overview of what is covered in the thesis.
- Chapter II describes the essential relevant literature pertaining to SI within the context of this thesis' landscape. This chapter includes the basic concepts and principles of key areas of knowledge (e.g. systems engineering, systems architecture, systems management, systems acquisition, etc.) related to SI elements working together in acquiring space-based systems.
- Chapter III provides an organized data collection and reduction from executing detailed steps of this thesis' two-part methodology with the intention of achieving the research's objective and answering the investigation questions.

- Chapter IV employs the traceability matrix with real-world space-system programs assessing SI attributes and demonstrating that SI can be traced and therefore be quantified.
- Chapter V summarizes the results and findings from the data analyses performed in Chapter III and from the case studies performed in Chapter IV. The chapter concludes with this thesis' contributions to this topic's body of knowledge to include recommendations for future researches.

II. Literature Review

The purpose of this chapter is to provide brief background information on topics areas used in the methodology and data analysis chapters. The intent is to frame the reader's mind in support of minimizing details on how the methodologies are conducted. Also, in situations where there are two or more literature references, this chapter explains which parts are being used by this thesis.

Defining Systems Integration

In search of a standard definition for SI within the DoD System Acquisition Lifecycle Framework, the authors initially found "System Integration" described as the first major effort of the System Development & Demonstration (SDD) Phase:

> "Systems Integration is intended to integrate subsystems, complete detailed design, and reduce system-level risk. The program shall enter System Integration when the PM has a technical solution for the system, but has not yet integrated the subsystems into a complete system." (38:11)

However, the updated DoD instruction has renamed this same effort as "Integrated System Design" of the Engineering & Manufacturing Development (EMD) Phase:

> "Integrated System Design is intended to define system and system-of-systems functionality and interfaces, complete hardware and software detailed design, and reduce system-level risk. Integrated System Design shall include the establishment of the product baseline for all configuration items." (37:21).

Oddly enough, the difference between these two definitions is focused on the definition of a "system." In order words, the obvious definition of SI as "working together" is not only dependent on the characteristics and attributes of a system but also the system's levels of abstraction.

This system-dependency was also discovered in searching for academic textbooks entitled with SI. Most of the textbooks were found within the realms of Systems Engineering, Systems Management, Systems Architecture and Enterprise Integration; except one surfaced with the title ‘System Integration’ (i.e., no “s” at the end of the word system). This published book corroborates the thesis’ basic SI model of people, process and product as shown Figure 1. The book describes “the work of many people from different functional disciplines, working on different system component products, in different process steps over time as the three fundamental integration components of function, product, and process respectively.” (21:42). Furthermore, the book extends the relationships between these integration components with “a three-valued situation: co-integration, cross-integration, and null values; creating a workspace of 27 (3 cubed) different integration possibilities.” (21:46). And yet, this thesis only addresses seven integration areas which are comparable to Grady’s cross-integration possibilities.

The rest of the textbooks (i.e., related to Systems Engineering, Systems Management and Systems Architecture) and some of the available organizational SE handbooks (i.e., INCOSE, NASA, SMC and MIL-STD) also back-up the definition of SI as the action of “working together” between the basic components of a system – people, processes and products. Although not explicit, their descriptions of SI activities can be correlated with at least one of the seven integration areas proposed in this thesis.

Systems Engineering and Integration

Systems Engineering and Systems Integration are often thought of a two separate disciplines and practiced independently. The size and complexity of modern DoD system acquisitions, most notably DoD space systems acquisitions, has prompted a serious re-evaluation of the roles these two disciplines play in an acquisition lifecycle. The DoD Acquisition Guidebook (DAG) describes Systems Engineering as interdisciplinary that requires the integration of numerous activities and processes.

> "Systems engineering is an interdisciplinary approach or a structured, disciplined, and documented technical effort to simultaneously design and develop systems products and processes to satisfy the needs of the customer. Systems engineering transforms needed operational capabilities into an integrated system design through concurrent consideration of *all* Lifecycle needs. As systems become larger and more complex, the design, development, and production of a system or system-of-systems require the integration of numerous activities and processes. Systems engineering is the approach to coordinate and integrate all acquisition Lifecycle activities. Systems engineering integrates diverse technical management processes to achieve an integrated systems design." (27:Ch 4, 3)

While this description affords Integration a place within Systems Engineering, it fails to provide Integration context within SE or to distinguish between the two domains (i.e., SI and SE). Leaving Systems Engineering as the overarching process includes elements of integration, subordinates SI to SE and discourages meaningful analysis of program SI.

Systems Engineering is inextricably linked to the process of Integration within the framework of DoD Systems Acquisition. This linkage, however, can be tenuous and highly dependent upon context. If a user's context is components and processes, the meaning of Integration might be:

> "Integration is the process of incorporating the lower-level system elements into a higher-level system element in the physical architecture." (27:Ch 4, 34)

> Integration is defined as the progressive linking and testing of system components to merge their functional and technical characteristics into a comprehensive, interoperable system. (35:Sec 4, 1-4)

> Integration means bringing things together so they work as a whole. System integration means bringing subsystems together to produce the desired result and ensure that the subsystems will interact to satisfy the customer's needs. (21:3)

When a user of the word "Integration" is working from the context of functions, the meaning of Integration might be:

> A cornerstone of the IPPD management technique is the integration of all stakeholders into a cohesive working unit. In traditional acquisition and development involving a sequential handoff of tasks, location of the various people was not a major concern. Today's IPPD approach makes real-time integration of a program's various functions essential. (26:7)

Even the DoD Integrated Product and Process Development (IPPD) Handbook does not attempt to define *Integration;* instead, throughout this document, the reader is admonished to ensure Integration in all things so that good will result. If integration is not achieved, the handbook warns, undesirable things will result.

Similarly, the Capability Maturity Model Integration (CMMI) standard does not attempt to define "Integration." CMMI doctrine tells us that "The organization is an integrated system capable of providing and sustaining the people, products, and processes necessary for the effective and efficient execution of its projects." The definition of *Integration* is conspicuously absent from the CMMI discussion of Integration and the reader is left to their own interpretations. (11:461)

> Life cycle integration is achieved through integrated development—that is, concurrent consideration of all life cycle needs during the development process. (53:7)

> Integration is obtained by designing a model or simulation to inter-operate with other models or simulations for the purpose of increased performance, cost benefit, or synergism. (53:122)

Space-Based System-of-Systems (SoS) Architecture

DoD Space System Acquisition programs increasingly face challenges of effectively making multiple elements work together as they strive to improve the delivery of also an increasingly technology-sophisticated solution for defense mission usage. Although Space System Acquisition programs have the same overall objectives as other defense acquisition programs, space-based system solutions differ in the sense that the orbiting space vehicle(s) must be designed and built to survive for years in the harsh space environment with no physical maintenance and a one-shot mission-assured deployment. The ground mission control systems are designed and built to operate these orbiting space vehicle(s) at a handful of locations. The launch vehicle is designed and built for one successful short-lived purpose, which is to send and deploy the space vehicle with specific payloads into its orbit. The Earth-bound users or subjects of these space-based systems are serviced several million miles away having to pass through a vast and harsh obstacle called the outer-space environment. Figure 2 illustrates the distinct elements of a typical space-based SoS showing that there is much work to be done in decomposing the integration work into fundamental parts and in putting together the results into a well-integrated, high quality, cost-effective overall system solution with complete user satisfaction in mind.

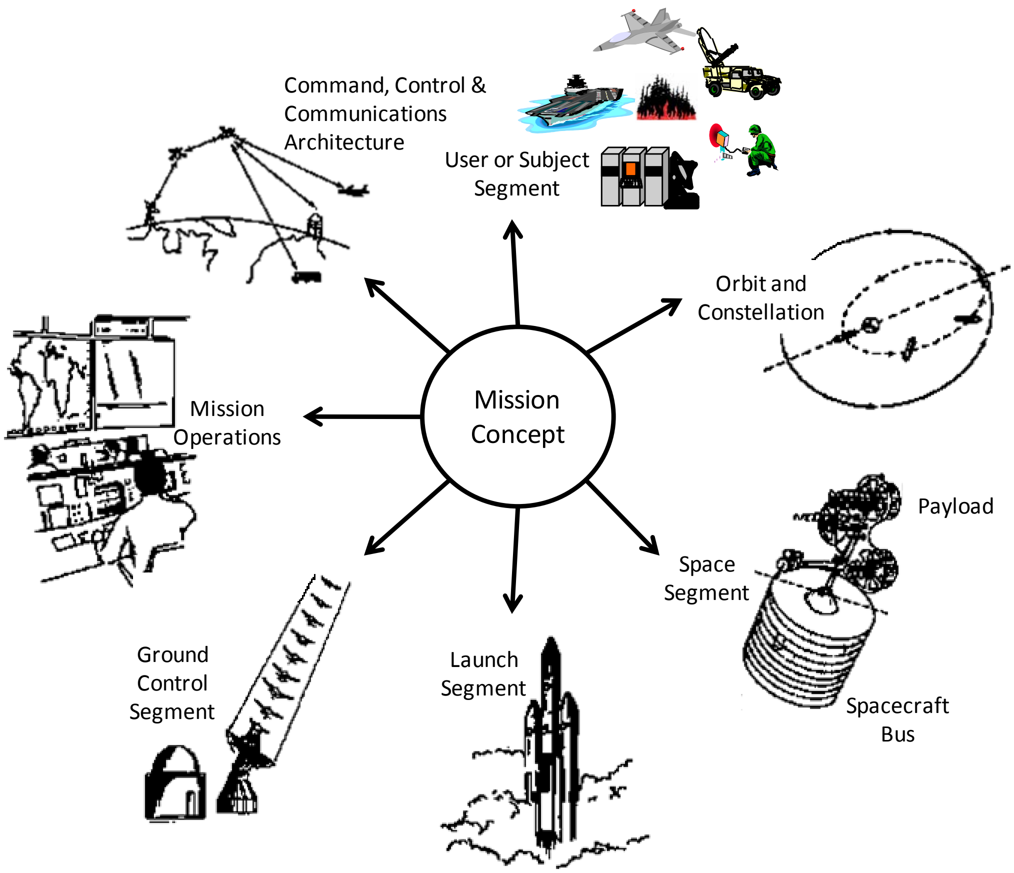

Figure 2. Space-Based System-of-Systems (SoS) Architecture (62:11)

DoD Space System Acquisition Lifecycle Framework

The framework used by this thesis to identify and trace SI activities is adapted from the National Security Space (NSS) Acquisition Policy (NSSAP) 03-01. The framework sets the lifecycle stage for decomposing the scope of work into sequential iterative phases to provide disciplined control. That is, the transition from one phase to another is usually defined by deliverables which are reviewed for completeness and accuracy and approved before work starts on the next phase. Figure 3 illustrates the five-phase lifecycle framework used to acquire space-based systems.

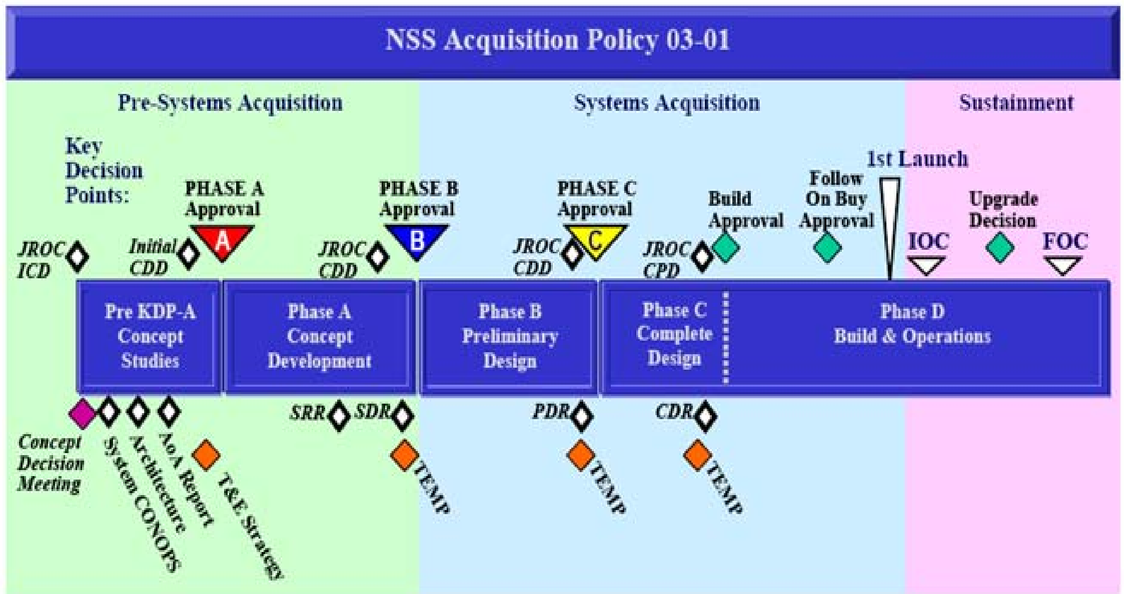

Figure 3. Space System Acquisition Framework (24:5)

Systems Engineering Technical Reviews & Audits

In pursuit of characterizing and how to trace SI, the research first explored the subject on Work Breakdown Structures (WBS) which "form the semantics frame of reference on the system levels of abstraction" (61:76). However, it was difficult to incorporate the phases (and their activities) of the Space System Acquisition Framework. Nonetheless, the research shifted to using TR&As as described in the NSSAP 03-01 and DoDI 5000.02. The later led the research to a more detailed illustration developed by the Department of the Navy as shown in Figure 4. The figure comes with a software tool that provides individual checklists for each TR&A, which provided a deeper understanding on characterizing and how to trace SI; and paved the path for the establishment of a methodology.

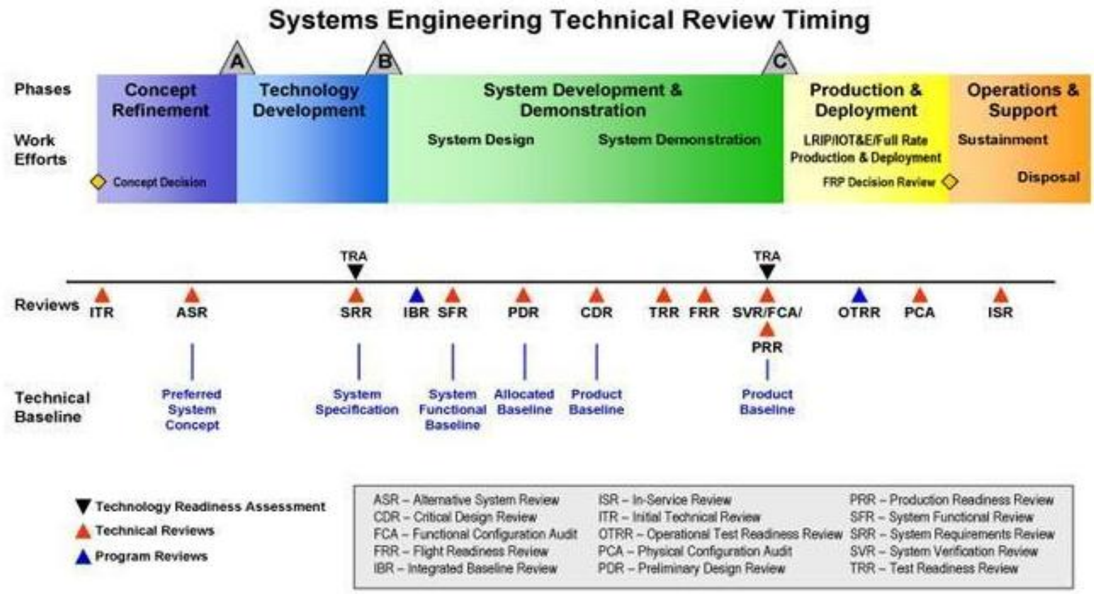

Figure 4. DoDI 5000.02 Systems Engineering Technical Review Timing (56:1)

Due to the lack of space-system terminology in the DoDI 5000.02 TR&A set, the research pursued with NSSAP 03-01 references which led to the Aerospace Corporation's Mission Assurance Guide (MAG) which help build the foundation of this paper's data collection. The following paragraphs are extracts that describe the relationship between lessons-learned and TR&A objectives.

> Technical reviews and audits entail a tremendous amount of detailed engineering and programmatic efforts. Not only do the reviews make it possible for the interfaces and composite performance to be understood, they also establish a schedule imperative with entrance and exit criteria that synchronize the government and contractor expectations. The reviews permit the MA experts to work in concert with program development resources and within the program's chain of command to fulfill their roles. (23:152)

> The review process also entails lesson learning. A Lesson Learned is understanding gained by experience—either positive (as in a successful

> test or mission), or negative (as in a mishap or failure). Sharing lessons from the NSS program—i.e., to identify, communicate, and record good practices and adverse experiences with implications broader than localized corrective actions—is an important MA mechanism that benefits the future work of the organization, especially in the prevention of recurrence of accidents. (23:152)

Table 1 maps the MAG's TR&As (includes traditional ones) with its Mission Assurance (MA) phase definition language into the acquisition phases referred to in the NSSAP 03-01 and the DoDI 5000.02 (23:23, 150-151). Appendix D provides brief descriptions of each TR&A to be used to populate the baseline Traceability Matrix-Model (Appendix C – Table 14).

Table 1. Mapping of TR&As and MA phases to NSS Acquisition Phases

<table>
<tr><th>TECHNICAL REVIEWS & AUDITS (TR&A)</th><th>MISSION ASSURANCE (MA) PHASE</th><th>NSSAP 03-01 PHASE</th><th>DODI 5000.02 PHASES</th></tr>
<tr><td><none></td><td>Phase 0: Concept Studies</td><td>Phase 0: Concept Studies</td><td>Material Solution Analysis Phase</td></tr>
<tr><td>Manufacturing Management/Production Capability Review (MM/PCR)</td><td rowspan="5">Phase A: Concept Development</td><td rowspan="5">Phase A: Concept Development</td><td rowspan="5">Technology Development Phase</td></tr>
<tr><td>Integrated Baseline Review (IBR)</td></tr>
<tr><td>System Requirements Review (SRR)</td></tr>
<tr><td>System Design Audit (SDA)</td></tr>
<tr><td>System Design Review (SDR)</td></tr>
<tr><td>Preliminary Design Audit (PDA)</td><td rowspan="2">Phase B: Preliminary Design</td><td rowspan="2">Phase B: Preliminary Design</td><td rowspan="4">Engineering and Manufacturing Development (EMD) Phase</td></tr>
<tr><td>Preliminary Design Review (PDR)</td></tr>
<tr><td>Critical Design Audit (CDA)</td><td rowspan="2">Phase C: Complete Design</td><td rowspan="2">Phase C: Complete Design</td></tr>
<tr><td>Critical Design Review (CDR)</td></tr>
<tr><td>Manufacturing/Production Readiness Review (PRR)</td><td rowspan="8">Phase D1: Fabrication and Integration</td><td rowspan="13">Phase D: Build & Operations</td><td rowspan="12">Production and Deployment Phase</td></tr>
<tr><td>Test Readiness Review (TRR)</td></tr>
<tr><td>Formal Qualification Review (FQR)</td></tr>
<tr><td>System Verification Review (SVR)</td></tr>
<tr><td>Hardware Acceptance Review (HAR)</td></tr>
<tr><td>Functional Configuration Audit (FCA)</td></tr>
<tr><td>Physical Configuration Audit (PCA)</td></tr>
<tr><td>Pre-Ship Review (PSR)</td></tr>
<tr><td>Independent Readiness Review Team (IRRT)/Mission Assurance Team (MAT)</td><td rowspan="4">Phase D2: Fielding and Checkout</td></tr>
<tr><td>Mission Readiness Review (MRR)</td></tr>
<tr><td>Flight Readiness Review (FRR)</td></tr>
<tr><td>Launch Readiness Review (LRR)</td></tr>
<tr><td>Post Flight Review (PFR)</td><td>Phase D3: Operations & Disposal</td><td>Operations & Support Phase</td></tr>
</table>

Space Systems Case Studies

A number of case studies involving space systems were reviewed for this thesis. Case studies, in general, were used to derive experiential data on systems that have the potential to employ significant Systems Engineering; and therefore, Systems Integration. From this data; the question of "How is Systems Integration Traceable?" will be answered. The lessons learned addressed by these case studies will aid in answering this question are as follows.

The Hubble Space Telescope (HST) Case Study

The Program did not integrate the user community into the acquisition process.

> "In the early stages of the HST program the mechanism for involving the customer was not well defined. The user community was initially polarized and not effectively engaged in program definition and advocacy. This eventually changed for the better, albeit driven heavily by external political and related national program initiatives." (31:vi)

> "Nonetheless, collaboration between government engineers, contractor engineers, as well as customers, must be well defined and exercised early on to overcome inevitable integration challenges and unforeseen events." (31:vi)

Life Cycle Support was integrated into HST early in program life.

> "Life Cycle Support planning and execution must be integral from day one, including concept and design phases." The results will speak for themselves. Programs structured with real life cycle performance as a design driver will be capable of performing in-service better, and will be capable of dealing with unforeseen events (even usage in unanticipated missions). (31:vii)

HST did not effectively integrate program risk management.

> "For complex programs, the number of players (government and contractor) demands that the program be structured to cope with high risk factors in many management and technical areas simultaneously." (31:vii)

HST provisions for a high degree of systems integration to assemble, test, deploy and operate the system is essential to success and must be identified as a fundamental program resource need from early on (part of the program baseline).

> "For HST, the early wedding of the program to the Shuttle, prior NASA (and of course, NASA contractors) experience with similarly complex programs, such as Apollo, and the early requirement for manned, on-orbit servicing made it hard not to recognize this was a big SE integration challenge. Nonetheless, collaboration between government engineers, contractor engineers, as well as customers, must be well defined and exercised early on to overcome inevitable integration challenges and unforeseen events." (31:7)

The Peacekeeper Intercontinental Ballistic Missile Case Study

Dating back to the Strategic Arms Limitation Treaty I (SALT I) under President Richard Nixon, the United States (US) was limited to a predetermined number of Intercontinental Ballistic Missiles (ICBMs). Increased ICBM capabilities had to come from quality and performance improvements in the ICBMs the US was authorized by the SALT I treaty to possess. "Missile X" (MX) was developed as a means to increase strategic combat capability through greater payload capacity and improved missile and warhead accuracy. Additionally, the U.S. wanted to greatly increase the survivability of its missiles which were becoming more vulnerable to improved Soviet missiles with very accurate warheads. The USAF began an acquisition program in 1973 known as "Missile X" and purposely leveraged much of the existing technologies of the successful Minuteman program. Under President Jimmy Carter, the program progressed and the missile entered full scale development in June 1979. On 22 November 1983, President Ronald Reagan directed that MX should be called Peacekeeper; thus the Peacekeeper program commenced.

Political and public turmoil over the introduction of another nuclear missile and some very expensive and resource-intensive basing modes associated with it obscured the success of this outstanding missile development program. The program's success was due primarily to a well-planned and executed systems development process involving highly complex and detailed Systems Engineering and Integration.

This case study, the authors attributed the relatively quick and successful space system acquisition program partially to the outstanding systems engineering environment that was created and nurtured at the Ballistic Missile Organization (BMO) during the 1970s and 1980s. Specifically, they point out the following key factors in the program's success:

- The disciplined systems engineering processes that were primarily developed by the Ballistic Missile Organization (BMO).
- The BMO staff that allowed systems engineering and program management staff to work multiple ICBM programs as part of a knowledge management process.
- The Associate Contractor Structure and the establishment of the USAF program office as the program integrator.
- The integrated use of the Systems Engineering/Technical Assistance Contractor (TRW) by the SPO to support program management and control the systems engineering process.
- The Systems Engineer had to be a leader and set the example in applying the engineering processes. That was fundamental. (50:iv)
- The Systems Engineer had to be a person who had good intuition in terms of how to approach and solve problems. The problems the Peacekeeper engineers encountered were similar to those experienced in developing the Atlas, Thor, Titan and Minuteman programs. They had different acronyms but they were the same kinds of problems. So they had to be able to invoke classical solutions to problems very quickly with little additional data. (50:61)

- The Systems Engineer had to ensure that the program was planned in detail, that the plan was realistic, and that people could actually implement that plan. (50:61)

- The Systems Engineer had to be cognizant of budget limitations, especially in a complex program like Peacekeeper, which entailed numerous budget lines. Engineers have an obligation to estimate costs in a comprehensive and complete manner. (50:61)

- The Systems Engineer had to be able to select competent people to be in charge of critical areas like propulsion, re-entry vehicle, launch control, and guidance. While the chief engineer is responsible, that person has to rely on the expertise of those on his design and development teams. (50:61)

- The Systems Engineer should not become involved in important program decisions without a complete understanding of appropriate technologies. (50:61)

- The SPO should manage the technology base for the next program so that lessons learned from the previous system can be applied and technology transition decisions can be made based on an intimate knowledge of the results of the technology development efforts.

- The case study also highlighted the importance of developing system engineers and allowing them to progress through a variety of missile programs to gain expertise. (50:61)

The Peacekeeper missile program provided a good example of a space system acquisition program that delivered a product near its projected budget and close to its scheduled IOC and FOC. The missile portion of the program was developed, produced, and managed in a manner that today's Air Force program offices would recognize. All technologies had demonstrated sufficient development and test before insertion in the missile. There was a rigorous systems engineering process and a reasonable test program. The missile program was adequately funded in a manner that caused few schedule perturbations. There were some production problems (motors and guidance systems), but those were addressed and fixed in a relatively short period of time. (50:62)

Global Positioning System (GPS) Case Study

GPS consists of three major segments: the Space Vehicle (SV), the User Equipment (UE), and the Control Station (CS). The space vehicle segment consists of a system of 24 satellites, configured in a constellation of six equally spaced orbital planes. Precise time is provided by a redundant system of rubidium and/or cesium atomic clocks on-board the SV. Each satellite is capable of continuously transmitting L1 and L2 signals for navigation and timing, and L3 signal for nuclear detonation sensor data. It is also capable of receiving commands and data from the master control station, and data from remote antennas via S-band transmissions (i.e., Air Force Satellite Control Network [AFSCN]). GPS has always been available to the civilian community, foreign and domestic.

The size, complexity, and importance of GPS dictate rigorous SI throughout all phases of its lifecycle. Achieving the desired level of SI proved to be an elusive goal; as key stakeholders continually changed, priorities were adjusted and technology increased exponentially. The number of groups needed to design subsystems, test these subsystems, and to integrate the subsystems eventually became so large and diverse that management of the groups became an endeavor tantamount to managing the program itself. This fact alone made SI paramount for the success of the GPS program.

> "Large aerospace companies have worked diligently to establish common systems engineering practices across their enterprises. However, because of the mega-trend of teaming in large (and some small) programs, these common practices must be understood and used beyond the enterprise and

> to multiple corporations. It is essential that the systems engineering process govern integration, balance, allocation, and verification, and be useful to the entire program team down to the design and interface level." (20:11)

> "The JPO decided to retain core systems engineering/system integration responsibility. Col. Parkinson had a concern with the potential for proliferation of systems engineering groups within an organization. He viewed systems engineering as a common-sense approach to creating an atmosphere to synthesize solutions based upon a requirements process, and to ensure good validation/verification of the design to meet those requirements. He advocated using good systems engineering principles to work issues as they arose." (20:38)

> "The integration role required contact with many government and industry entities. A plethora of technical expertise organizations, test organizations, users, etc. required working interfaces and integration." (20:39)

During this time frame, the GPS JPO Director determined that the Systems Engineering Directorate needed to assume a more aggressive integration role. He believed that unresolved issues between the GPS segments and/or systems were inappropriately being channeled to his level for resolution. He wanted the Systems Engineering Directorate to take on the responsibility for the integration between the system segments. By doing so, it was felt, SI would be equitably distributed among the GPS segments and more tightly controlled by the management hierarchy best suited to accomplish the tasks. Conflict resolution between these segments would then only filter up to the GPS JPO when all attempts at lower levels had failed.

Other SE&I Relevant Subject Areas

Systems Integration is a very broad application on the user's definition, scope, and viewpoint of the system. While pursuing and achieving this paper's objectives, the following relevant topics help identify and trace the characteristics and attributes of SI and provide support evidence in defense of this thesis. Each topic briefly describes its structure, function and purpose.

Concurrent Engineering

During the search of SI literature, Concurrent Engineering mostly came up as an alternative Systems Engineering approach to integrating engineering specialties to improve the product development process. This approach attempts to integrate functional disciplines such as supportability, manufacturing, assembly, quality control, finance, marketing, and customer service during the design phase of the system's life cycle framework. The specific description of concurrent engineering as a "systems-engineering perspective that focuses on integrating people, processes, problem-solving mechanism, and information" (42:328) supported this research's SI model as shown in Figure 1. In addition to this academic reference, the National Aeronautics & Space Administration's (NASA) Systems Engineering Handbook implements this approach in the early stages of its Program/Project Lifecycle process flow (46:34).

Systems Re-Engineering

Another SE (or rather SI) approach designed to integrate changes to a system solution already in service use. In periods of high-velocity environments where continual organizational change and associated change in processes and product occur, re-engineering is employed "at the level of systems management, process, product, or any combination of these" (42:827). A whole different subject matter all together, reviewing Systems Re-Engineering literature supported this thesis' SI-model as described in Figure 1.

<u>Human Systems Integration (HSI)</u>

This subject matter also came out a lot in the search of SI. Clearly, the topic's name inclusive of SI implies some relevant relationship. Looking closely at the subject matter, Human Systems Integration indeed implicates actions of working together. In this case, human factors are integrated into the design of a system solution. Another name associated to HSI is Human Factors Engineering (HFE). This design for usability approach supports this thesis' idea that people is part of a complete system – that is, addressing the human being and the interfaces between the human and other system elements (5:459).

<u>Enterprise Systems Integration</u>

A newer application of SI in today's environment of increasing demands for a system of systems that brings people around the world closer to each other at a touch of a button. "The wave of corporate mergers and growth of businesses using the Internet have boosted enterprise systems integration's profile in both Information Technology (IT) and business management. Integrated enterprise systems can provide information across all points in an organization and the unique way systems integration blends business practices and IT" (34:xv).

Literature Review Summary

A review of the core DoD literature devoted to SE and SI revealed that these two areas of engineering are inextricably linked within the framework of DoD Systems Acquisition. This linkage, however, can be tenuous and highly dependent upon context. Although discussions of SE and SI are found throughout the core DoD engineering literature and space systems case studies, there is a complete absence of the formal, structured relationship between the two engineering disciplines. Based on this literature review, it is concluded that a method for tracing SI in a Space System acquisition has not yet been proposed and documented.

III. Methodology

Considering the lack of a clear definition and standards for SI, this chapter will describe a step-wise path for identifying the characteristics of SI and tracing the value of SI within a space-system acquisition. It will detail how data sources are selected and how the resultant data can be collected and reduced to support capturing measurable SI activities. The methodology development approach begins with gleaning SI concepts from space-system acquisition lessons learned. These SI concepts will then be compared to the products derived from the SI model based on integrating people, processes and products. The methodology development path proceeds on by linking the objects in the seven SI areas and their interdependent roles in the technical reviews and audits processes; all occurring throughout the acquisition lifecycle framework.

Identifying Systems Integration Characteristics

The first step of this research's methodology involves characterizing SI to build a foundation on to which to develop logical, consistent and standardized (within this thesis) measures intended to assess space-system acquisition activities. The idea is to employ the proposed SI model (as Chapter 1 describes) in capturing SI descriptions that uniquely identifies and distinguishes itself from its obvious implication. The intent is to develop a foundation in understanding how to characterize SI in terms of its attributes, properties, and performance. In the course of doing so, the concept of the proposed SI model (having seven areas where SI can occur by *interacting*, *interrelating*, and *interfacing* the basic system elements of *people*, *processes* and *products*) is validated.

Irrespective of the disappointing outcome of finding little to no literature on 'what

SI is and its value within DoD systems acquisition', it is evident that there is substantial documentation where SI characteristics can be derived or captured. On account of these great volumes of text to make known SI attributes, properties and characteristics; content analysis is the research tool chosen as best summarized by the following citation:

> "Content analysis is a research tool used to determine the presence of certain words or concepts within texts or sets of texts. Researchers quantify and analyze the presence, meanings and relationships of such words and concepts, then make inferences about the messages within the texts, the writer(s), the audience, … To conduct a content analysis on any such text, the text is coded, or broken down, into manageable categories on a variety of levels--word, word sense, phrase, sentence, or theme--and then examined…to code for existence or frequency…" (10:1).

The tool outputs a frequency tally of conceptual words and phrases inferred from these texts of which can be used with traditional quantitative statistical methods. However, this research only uses the method to organize a qualitative assessment of SI inferences.

Selecting the data source for this part starts with existing space-system related case study reports. The intent is to extract text that infers any *interacting*, *interrelating*, and *interfacing* actions between or among *people*, *processes* and *products*. This process required an abundance of data collection efforts, re-enforced with a background of knowledge on how space-systems are engineered and acquired in the DoD. During the search, researchers found a collection (13:1-100) of 100 recognized space-system root-cause-problem reports (PRs) investigated during the years of 1985-2005 with an average of 4.6 lesson-learned (LL) statements per PR. A total of 465 lesson-learned statements resulted, providing already prepared, fact-based root-cause-problem analyses performed by space-system acquisition experts from the Aerospace Corporation.

Data collection starts with a structured working document that lists LL statements for each PR, while leaving room for identifying SI-elements and SI-area inferences. Each LL statement is analyzed for conceptual texts which are italicized and translated into the SI-elements of people, process or product. At the same time, these texts are linked together inducing relationships that explain the actions of "working together" as represented by the seven SI areas. The most of what can be induced from this approach is "this element may be integrated with the other element." However shallow, the underlying logic allows deeper semantics in the translation. The collection is imported to a spreadsheet (using Microsoft™ Excel) for ease of tallying recurrences.

Data reduction is minimally done since the LL statements were prepared for formal reporting. Each LL report "tells a failure story, spotlighting three questions: How did the mistake occur? What prevented its detection? Why did it bring down the entire system?" (10:21). Therefore, none of the LL statements were altered. The sentence structure, the words, and the conceptual content are kept intact. Touch-up efforts are mostly required due to "cut-and-paste" mis-translations.

Chapter IV describes how the collected and reduced data is coded and analyzed to include the findings and implications along with the interpretations of the results.

Tracing Systems Integration Characteristics

The development of the traceability methodology continues by tracing SI characteristics as evidence to its relative value in the space-system lifecycle acquisition framework. The significance of tracing SI is two-fold. First, the ability to trace SI through an acquisition program provides the Program Manager (PM) and other stakeholders with insight into the efficacy of their SE endeavor. Program SE consumes a significant percentage of overall program budget and schedule and therefore must be as efficient and effective as possible. Second, SE is often arcane and details can elude even the most astute PMs. Thus, tracing SI provides the PM with valuable knowledge to support program implementation and reveal progress of the SE being conducted to achieve program objectives.

Although SI is often perceived as intuitive, forward traceability analysis must be used to link the proposed SI elements and areas from one lifecycle phase to the next. The intent is to establish a model in the form of a checklist-like matrix to help validate the interrelationships, interactions, and interfaces of people, processes, and products that are evident in the objectives, criteria and artifacts of a TR&A event. This forward tracing uses a series of TR&As that are timed to the acquisition lifecycle phases and imply a multi-level system solution from concept to operations. These TR&As are not only necessary for advancing through Key/Milestone Decision Points (KDP/MS) and acquisition phases, but also for performing the Systems Engineering and Integration (SE&I) necessary for achieving system life cycle cost, schedule and performance thresholds. With these inherent aspects in TR&As, SI characteristics are traced throughout the lifecycle framework and are mapped from inputs to outputs of a multi-level system.

The development of the traceability matrix-model starts with gathering the appropriate information for the trace. Results from the context analysis of the LL statements form the information that feeds into this matrix-model. The high frequency tallies of SI areas are listed in the matrix as “row” items. Whereas, the TR&A set is listed in the matrix as "column" items. The intersection between the row and column constitutes a "checkbox" to indicate the presence of the SI area in that TR&A event. Table 2 illustrates the basic outline of this matrix-model to be populated with SI areas identified from the LL statements and the choices of TR&A events.

Table 2. Tracing SI Matrix-Model

		Phase 1	Review 1	Phase 2	Review 2	Review 3	Phase 3	Review 4	Phase 4	Review 5	Phase 5	Review 6	Audit 1
A: People-People													
B: Process-Process													
C: Product-Product													
D: People-Process													
E: Process-Product													
F: Product-People													
G: People-Process-Product													

Appendix C of this thesis establishes the traceability matrix-model (Table 14), which will be populated with checkmarks linking these SI area-combinations with the entrance and exit criteria described in the recommended TR&A literature and their brief descriptions in Appendix D. Finally, the number of SI areas found in each TR&A event forms the baseline to gauge the importance of SI in successfully accomplishing the tasks required by a TR&A. These frequency numbers are not intended to be a key measure to determine a pass/fail criterion for the individual TR&A, or for that matter the overall program SI. The intention of these frequency numbers is to provide a context baseline for assessing the SI traceability of a space-system program as it transitions from one TR&A to another. By doing so, a member of the program management team can demonstrate that SI is flowing-down the multi-level system and transitioning across the lifecycle phases. However, this demonstration does not guarantee adequate program SI, but rather indicates knowledge of past space-system SI challenges and willingness to overcome these challenges.

This traceability matrix-model can be used with real-world space-system acquisition programs; which would then fully validate its general application. The programs' TR&A data is assessed for the presence of these SI areas. A search for similar functionality names and concepts is conducted when making a trace. When correlated, the "checkbox" is marked, validating that this particular SI activity occurs during that specific TR&A event. To conclude, the tallies of the number of SI areas found in each TR&A event are compared to the model's numbers; demonstrating TR&A traceability only acknowledges proof that a program has fulfilled the SI intent of that TR&A transition.

As an example of the traceability matrix-model methodology, the GPS Case Study was subjected to the aforementioned process. The results of this process are shown in Appendix C – Table 15.

IV. Analysis and Results

Using the outputs from the data collection and reduction described in the methodology section (i.e. Chapter III), this chapter records the SI characteristics induced from the identification approach to include any findings that would help better understand what SI is about and the value of SI in space-system acquisition. Furthermore, the traceability matrix model is implemented using the GPS Systems Engineering Case Study (20:1-72). The intent of this implementation is to validate the utilization of the matrix-model as a demonstration of improved efficiency, effectiveness and timely delivery of defense systems in general.

Characterizing Systems Integration

The main focus of the identification step is to infer from an LL statement the action(s) of the SI elements (i.e. people, processes, and/or products) "working together". The interrelating, interacting or interfacing linking relationships between/among the inferred concepts are categorized into fitting integration areas as described by the proposed integration areas presented in Table 3. Recognizing that interpretation of what constitutes an SI area might be biased due to the known body of knowledge at the time of statement analysis. More often than some, going through more statement analysis widens the analyst's mind to ideas that was not within the criteria of the previous examination.

Table 3. Identification of Integration Areas

ID	DESCRIPTION	DEFINITION
A	People-People	Person/People interrelating with other Person/People. Example: satellite contractor collaborating with launch contractor
B	Process-Process	Process(es) interacting with other Process(es). Example: design for test
C	Product-Product	Product(s) interfacing with other Product(s). Example: hardware functioning with software
D	People-Process or Process-People	Person/People integrating Process(es) or Process(es) integrated by Person/People. Example: engineers analyze or evaluate assuring operators
E	Process-Product or Product-Process	Process(es) integrating Product(s) or Product(s) integrated by Process(es). Example: test requirement* or subsystem undergoes test
F	Product-People or People-Product	Product(s) integrated by Person/People or Person/People integrating Product(s). Example: requirement* driven by operator's need or program office dealing with proprietary data
G	People-Process-Product	Person/People integrating Process(es) that integrates Product(s). Example: satellite contractor models launch vehicle

The following description demonstrates the identification step with three actual randomly selected LL statements. In each statement, concepts that fit a characteristic of an SI element are italicized. The details surrounding the chosen "concepts" in a statement and the results of the author's root-cause-problem analysis help with the character analysis of the fundamental action(s) of SI. The approach is simply identifying SI elements and actions that may be characterized appropriately into one or more integrated area(s) as previously described in Table 3. The integration area(s) with the related "italicized concept(s)" that represents the SI element(s) are listed below the lesson-learned statement. An asterisk (*) following a concept word indicates a requirement, which this thesis regards as a *product* item. Examples:

- Many *programs* require an independent *analysis* to ensure correct *modeling*.

  People-Process: program – analysis

  Process-Process: analysis – modeling

- Implement exception *handling* to *protect* the flight *processor* from aborts due to *data* handling errors.

  Product-Product: protect* – processor – data

  Process-Product: handling – data

- *Review* and *follow* operating and transportation *procedures* associated with cryogenic *equipment* to ensure *safety* to *personnel*, flight *hardware*, or *facilities*.

  Process-Process: review – follow – safety – facilities

  People-Process-Product: personnel – safety – procedures*

  Product-Product: procedures* – equipment – hardware

Summarizing, there are two “B” (Process-Process), two “C” (Product-Product), one “D” (People-Process), one “E” (Process-Product), one “G” (People-Process-Product) and the rest of the integrated areas are not identified.

This content analysis starts with a compendium of lesson-learned statements from the Aerospace Corporation technical report on space systems acquisition. There are 100 root-cause-problem reports (PR), each having an average of 4.6 lesson-learned (LL) statements, totaling to 465. For each LL statement, there is at least one involved SI area, totaling to 852. Although the LL statements revolve around space-system acquisition and engineering terminology, data coding is necessary for some concepts that were specifically chosen for emphasis; but can still be translated to a common term. Appendix A captures these LL statements for each PR title; to include the researcher's categorization of the involved SI areas. Behind the scene is a spreadsheet to tally the number of identified SI areas determining the rate of recurrence of each area. Figure 5 provides a graphical scorecard of the tallied seven SI areas found in these LL statements.

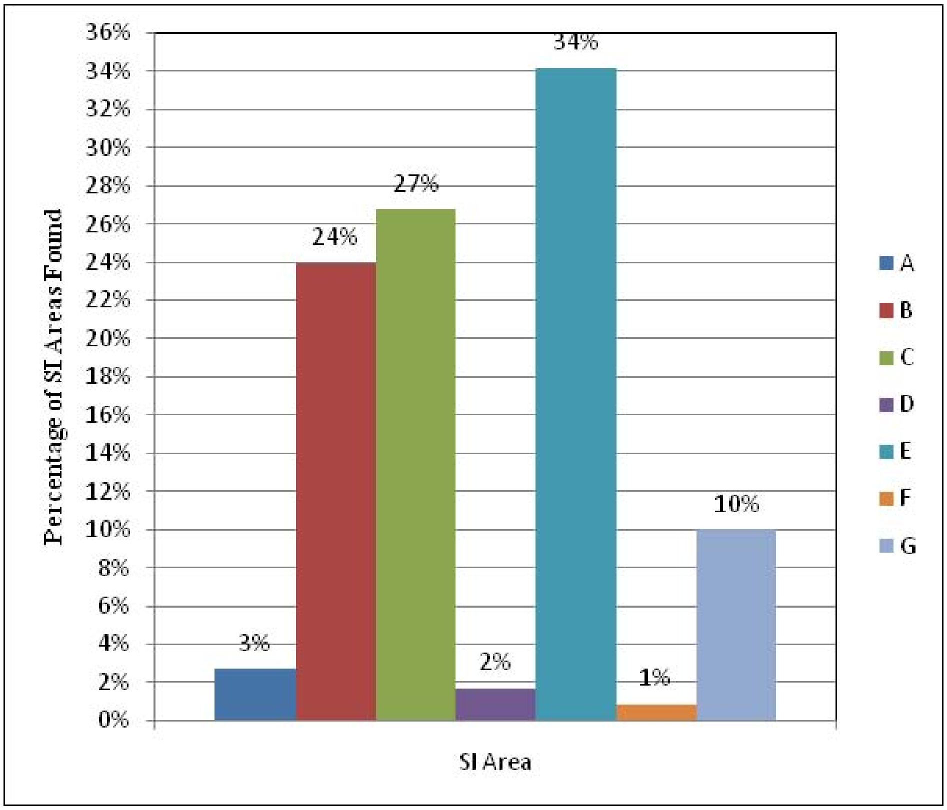

Figure 5. Systems Integration Areas Found in Lesson-Learned Statements

Continuing with the identification process, these LL statements are analyzed by its "face value" and decoded to bring about concepts that reflect characteristics related to SI. That is, the concepts used in a statement are examined for fitting characteristics to the SI element of people, process or product. Each LL statement is a point of a root-cause analysis emphasizing an effect that is related to a space-system acquisition problem.

Appendix B provides the coding sheets for each SI element. The coding sheets record conversions of the "concepts" extracted from the LL statements to traditional or commonly used words in space-system acquisition activities. For example, the words of "checkout, validate, verify, inspect, and test" are coded to the process of "evaluate". Table 4 capturers the top-level, type-labels for people, process or product used in space-system programs; whereas, Table 5 briefly defines each of these SI element types with respect to space-system acquisition programs.

Table 4. Common Types of Top-Level SI Elements in Space-System Programs

PEOPLE	PROCESSES	PRODUCTS
Acquirer	Define	System
Developer	Design	Segment
Operator	Implement	Subsystem
Stakeholder	Evaluate	Hardware
Supplier	Deploy	Software
User	Manage	Data
		Requirements

Table 5. Space-system Acquisition Program SI Element-Type Definitions

SI ELEMENT-TYPE	DEFINITION
Acquirer	People who are overarching of the management and financial for delivering an efficient, effective and suitable system solution to the war-fighter for mission use.
Developer	People who engage with the architecture, design, fabrication, coding, assembly, and production of the system solution.
Operator	People who "fly" (i.e. command and control) the spacecraft ensuring it "stays within its box."
Stakeholder	People who have interest on benefiting from the program or who are affected by the operations of the system entity.
Supplier	People who are outside of the Developer's organization that provides parts of the system entity to include within scope services to the Developer.
User	People who utilize the payload of the spacecraft. For example, a communications terminal user uses the comm.-payload to communication from one location to another; or a positional device user uses the navigational-payload to get data for the whereabouts.
Define	A top-level process that encompasses phase-activities related to identifying and analyzing the system entity, system architecture, system mission, system operations, system capability and system concept synthesis.
Design	A top-level process that encompasses phase-activities related to allocating the definitions as requirements into configurable, manageable items.
Implement	A top-level process that encompasses development phase-activities like fabrication, code, assembly, production and modification of the physical design.
Evaluate	A top-level process that encompasses phase-activities related to test, validation, verification, and quality assurance of a measurable item.
Deploy	A top-level process that encompasses phase-activities related to launch, install, flight, operations, and support process-activities.
Manage	A top-level process that encompasses the planning, scheduling, and budgeting for the program.
System	Or system-of-systems (SoS) represents the level of abstraction that describes the top-level representation of the entire solution from a user’s or operator’s frame of reference. (61:81)
Segment	Refers to system entities at the first level of decomposition below the SYSTEM level. (61:81)
Subsystem	Refers to system entities decomposed below the SEGMENT level.

SI ELEMENT-TYPE	DEFINITION
Hardware	A product made of material (i.e. physical) and its components (mechanical, electrical, electronic, hydraulic, pneumatic). (36:5)
Software	A product composed of a set of computer programs, procedures, and possibly associated documentation and data. (36:6)
Data	Recorded information of any nature (including administrative, managerial, financial, and technical) regardless of medium or characteristics. (36:4)
Requirements	Any condition, characteristics, or capability that must be achieved and is essential to the end item's ability to perform its mission in the environment in which it must operate. (61:316). In this thesis, this product element includes formal documentation like specifications, plans, and manuals.

SI Element Coding

For ease of sorting and frequency tallying, the "concepts" extracted from the LL statements are individually captured and entered into a spreadsheet. After the variety of concepts collected from a LL have been assessed for intended meanings; these concepts will be grouped and assigned a single code word that represents the composite intended meaning. The frequencies of occurrence for each coded-concept are then tallied; indicating its relative importance to the SI element. The coding sheets for the people-element in Table 11, process-element in Table 12, and product-element in Table 13 are provided in Appendix B. Table 6 provides the frequency tally of coded-item-types of SI elements to be integrated.

Table 6. Frequency Tallied (Coded) Types of SI Elements

PEOPLE	**#**	**PROCESSES**	**#**	**PRODUCTS**	**#**
Acquirer	18	Define	82	System or SoS	46
Developer	106	Design	135	Segment	75
Operator	15	Implement	125	Subsystem	38
Stakeholder	9	Evaluate	296	Hardware	544
Supplier	10	Deploy	213	Software	85
User	1	Manage	84	Data	61
				Requirement	229
TOTAL	**159**	**TOTAL**	**935**	**TOTAL**	**1078**
GRAND TOTAL = 2172 SI element-instances					

Organizing these LL statement concepts into the top-level types of SI elements required insight on the lower level efforts of each element. The challenge is accurately rolling-up the details into broader concepts without losing any pertinent information in support of a high-confident assessment. Due to its 159 instances (i.e., the lowest tally of the three SI Elements) and less use of synonymic concept choices, the *people* SI element's coding required few retrospectives of the lesson-learned report. Coding the *process* SI element of 935 instances into top-level definitions is a bit more challenging because of having a broader synonymic spectrum of contextual choices. Despite the direct representation and unique character of a *product*, this SI element of 1078 instances is the most challenging because the items that are called-out span across all system levels of abstraction from parts to the system.

SI Area Coding

Using the same code sheets, each SI area underwent the same translation process. Extra effort on organizing the concepts the same way was required to utilize the spreadsheet's capability to sort and support ease of tallying the SI area combination. The same underlying assumption of having a higher frequency number indicates the importance or value of the SI area. These combinations are used in establishing the traceability matrix-model showing the areas that need more attention than the others in a particular TR&A event. Table 7 lists the first highly-tallied 10 combinations with their frequency (#) for each SI-area; whereas the full SI-area tallies are shown in Appendix C (Table 14).

Table 7. First 10 Highly-Tallied (Coded) SI-Area Combinations

#	A: People – People	#	D: People – Process
10	Acquirer – Developer	3	Developer – Evaluate
4	Developer (Launcher) – Developer (Satellite)	2	Developer – Define
3	Developer – Supplier	2	Developer – Design
2	Developer (Payload) – Developer (Bus)	2	Developer – Implement
1	Acquirer – Stakeholder (Independent)	2	Developer – Manage
1	Developer – Developer	1	Acquirer – Define
1	Developer – Operator	1	Acquirer – Manage
1	Developer (Sub) – Developer (Prime)	1	Operator – Define
1	Developer (System) – Developer (Software)	1	Stakeholder (Independent):Define
1	Stakeholder (AFSPC) – Stakeholder (Public)		
#	**B: Process – Process**	**#**	**E: Process – Product**
26	Evaluate – Deploy	40	Deploy – Hardware
17	Design – Evaluate	38	Evaluate – Hardware
13	Define – Evaluate	24	Design – Hardware
13	Design – Deploy	20	Evaluate – Requirement
13	Evaluate – Evaluate	10	Design – Requirement
13	Implement – Evaluate	10	Implement – Hardware
12	Deploy – Deploy	10	Manage – Requirement
9	Deploy –Manage	9	Define – Requirement
8	Evaluate – Manage	9	Deploy – Requirement
8	Implement – Deploy	6	Deployment – Segment
#	**C: Product – Product**	**#**	**F: People – Product**
62	Hardware – Hardware	2	Supplier – Requirements
38	Hardware – Requirement	1	Acquirer – Data

17	Segment – Hardware	1	Developer – Data
13	System – Hardware	1	Developer – Requirements
12	Hardware – Software	1	Operator – System
9	Hardware – Data	1	Stakeholder – Hardware
7	Segment – Requirement		
5	Hardware – Software – Requirement		
5	Requirement – Requirement		
5	Software – Requirement		
#	**G: People – Process – Product**		
7	Developer – Evaluate – Hardware		
7	Developer – Implement – Hardware		
4	Developer – Design – Hardware		
4	Developer – Manage – Requirement		
3	Developer –Define – Requirement		
3	Developer – Deploy – Hardware		
3	Developer – Design – Requirement		
3	Developer – Evaluate – Requirement		
3	Operator – Deploy – Hardware		
2	Developer – Deploy – Requirement		

Findings – SI Characteristics

This section reports what was discovered about SI from the lesson-learned statements by inducing actions between or among the basic systems elements of people, process and product. The findings indicate that SI is indeed obvious and often taken for granted. Of all the SI element instances combined (i.e., 2172), the concept of integration or any of its derivatives was mentioned three times as once a *people* and twice a *process* element – system integrator and integrate, respectively. Despite the non-verbal or non-written implications of SI in the construct of the statements, there are apparent cause-and-effect interactions, interrelations and interfaces within the elements of a lesson-learned.

<u>‘PRODUCT’ Oriented</u>

Results of 1078 “product” instances from the context analysis confirm SI’s ‘grass roots’. The astounding numbers on the *product* SI element and SI area “C” (that involves only itself) corroborates the historical background of SI being confined to the technical aspects of developing systems. That is, SI has been a part of the broad area of systems engineering. The 'Hardware' tally of 544 (about half of the 'product' instances) confirm that SI has a physical connotation and piecemeal quality – making different pieces of equipment work together. Next in line is 'Requirement' with 229 instances rightfully so supportive of the different pieces. The rest of the product-element-types are about equally important to the SI picture.

Rolling-up to higher levels in the product system hierarchy, 'Segment' is emphasized between the subsystem, segment and system (i.e. 38, 75, and 46 respectively) element-type. Oddly enough, this is the level where the people-element in each segment is represented by different organizations – that is, contracts in space-system acquisition language.

The first 10 high frequency tally of product-element combinations (as shown in Table 7) with the other elements are represented in the following SI-areas:

- "<u>Product-Product</u>" (C) has a high tally of 62 instances where 'Hardware' is interfacing with other 'Hardware' and a total tally of 67 instances with other product-items – 'Data, Requirement and Software', excluding the system hierarchy product-items. The 'Hardware' product-item does interface within the 3-system levels (as decomposed in this thesis) – system, segment, subsystem. The results confirm the main feature is the physical characteristic of SI represented by 'Hardware'.
- "Process-<u>Product</u>" (E) has 'Evaluate-Hardware' as the top in list with 48 instances; 'Deploy-Hardware' with 40; 'Design-Hardware' with 31; 'Implement-Hardware with 12; 'Manage-Hardware' with 7; and 'Define-Hardware' with 6. For the second

product-element contender ‘Requirement’, the combining process-element ‘Evaluate’ is first in the queue; followed by ‘Design, Manage, Define, Deployment, and Implement’ in ranking-order. This result emphasizes the important relationship between SI and test activities during product development.

- “People-<u>Product</u>” (F) has ‘Requirement’ as most significant product-element combined equally to all people-types – ‘Acquirer, Developer, Operator, Supplier, and Stakeholder’. This finding shows requirement starts and ends with the people’s needs.
- “People-Process-<u>Product</u>” (G) has all product-types evenly distributed among the other SI-elements. Despite the fairly even distribution, the ‘Hardware’ product-type can be seen as the main feature by some group-tally spikes.

This mastery in product integration has indeed been demonstrated by the many large, interoperable, and complex space systems being used today – civil and military alike.

<u>‘PROCESS’ Centered</u>

Following closely behind is the *process* SI element with 935 instances. This revelation is not a surprise due to the fact that these “integrated” processes brought about the product system. The adequacy of these processes working together is reflected by the performance of the product system they created. The number of SI area “Process-Process” (B) combinations indicates the need for coordination and control – an integrated management system (21:7). Thus in some SE circles of discussion, the integrated action between Systems Engineering and Program Management came into being – Systems

Engineering Management (42:113).

Looking closely at the tallies of the element-types, ‘evaluation’ with 296 instances comes at the top of the list. In this thesis’ context, the process-type ‘Evaluate’ encompasses any activities or variations related to test that is being performed at any in-depth degree and level in the system hierarchy. The element-type ‘Deploy’ comes in second with 213 instances. The literature review of SI reveals that the obvious part of working together can be seen in action in both these process-element types. Descriptions of integration & test are commonly found; whereas portions of ‘Deploy’ as defined in this thesis are usually described as Integrated Logistics Support (ILS).

The process-element types of ‘Design’ (135) and ‘Implement’ (125) are following behind. Both processes usually are not explicitly described with SI compared to the first two processes. Nonetheless, these processes are performed in multi-level actions of working together – systems integration in vertical motion of the hierarchy. The rest of the process-element types, ‘Define’ and ‘Manage’, are just as to the integration picture. Having a low number for ‘Define’ perhaps implies that the coding for product-type ‘Requirement’ should have been applied to the process-element.

The first 10 high frequency tally of process-element combinations (as shown in Table 7) with the other elements are represented in the following SI-areas:

- “<u>Process-Process</u>” (B) has ‘Evaluate-Deploy’ as the top in the list with 26 instances; and the remaining ‘Evaluate’ combinations with ‘Define, Design, Evaluate (different levels), and Implement’ have fairly equal tally of about 13 instances each. This result confirms the tightly coupled relationship of SI and test activities.

- "People-<u>Process</u>" (D) shows for every process-type 'Define, Design, Implement, Evaluate, Deploy and Manage' the interaction of all the people-type 'Acquirer, Developer, Operator, Supplier, and Stakeholder' with some fashion or other. Involvement of these people-types in any of the process-types is supporting of successful SI foretold by the lessons learned.
- "<u>Process</u>-Product" (E)'s first 3 combinations of 'Deploy-Hardware', 'Evaluate-Hardware', and 'Design-Hardware' have 40, 38, 24 instances; respectively. In these tallies, the process-elements do not make the combinations come to the top but rather their product-element 'Hardware'. These combinations just show that the 'Hardware' needs SI attention the most during 'Deploy, Evaluate and Design'. Looking closely to the next group of combinations in this SI-area, the product-element 'Requirement' comes next.
- "People-<u>Process</u>-Product" (G) shows 'Evaluate, Implement, and Design' as the process-elements of the first 3 combinations. Again, the product-element 'Hardware' is the focus of this SI-area.

Today's successful utilization of space systems demonstrates this mastery of SI skills in coordinating and controlling (i.e. managing) the overall technical direction of their development.

<u>'PEOPLE' Driven</u>

The tally of people SI elements resulted to a very low number (i.e. 159 instances) compared to the other two elements (i.e. process – 935 and product – 1078 instances). Seemingly, the low number suggests that the people involved in these lessons-learned do a good job working together; whereas the people who were pointed out are challenging to work with. However it may seem, the people who are a part of this SI system drives and accomplishes the interactions, interrelationships and interfaces between and among these SI-elements.

The 'Developer' with 109 instances leaves behind all the people-types. Understandingly, the developer is mostly contracted to "integrate" the process-types resulting into the product-types. The rest of the people-types are the recipients who may discover this integrated system as useful, effective, and affordable.

The first 10 high frequency tally of people-element combinations (as shown in Table 7) with the other elements are represented in the following SI-areas:

- "<u>People-People</u>" (A) has 'Acquirer-Developer' as the top in the list with 10 instances; 'Developer-Developer' with 6 each developing different portion of the overall system solution; and 'Developer-Supplier' with 3 showing interactions within same team. The rest of the combinations have 1 instance each. The results confirm the main driver of most of these processes – the Developer.
- "<u>People</u>-Process" (D) has 'Developer-Evaluate' as the top with 3 instances; 'Developer' interacting with 'Define, Design, Implement and Manage' has 2 instances each; and the rest with one each – 'Acquirer-Define', 'Acquirer-Manage', 'Operator-Define' and 'Stakeholder-Define'. Involvement of Acquirer, Operator and Stakeholder in the 'Define' process is very desirable as confirmed by the lessons learned.
- "<u>People</u>-Product" (F) has explicitly 'Supplier-Requirement" as the top with 2

instances; ‘Developer’ with ‘Requirement’ and ‘Data’ having 1 each; and the rest having one each – ‘Acquirer-Data’, ‘Operator-System’, and ‘Stakeholder-Hardware’. Results still corroborating ‘Developer’ as main driver, whereas the rest have supporting roles.

- “People-Process-Product” (G) shows a high tally of 60 instances where the ‘Developer’ with a variety of combinations is driving different processes to produce different product-items. The ‘Operator’ combinations of 11 emphasize the operator’s role as recipients to the products produced by the driven processes.

This mastery of people SI skills is indeed demonstrated by the successful performance of the space systems being used today.

‘PEOPLE–PROCESS–PRODUCT’ Integrated

Isolating the characteristics of each SI element made evident the need to accomplish the action of all three working together. The people system drives the process system to bring into being the product system with value to the customer. However, the results of the data collection as shown in Figure 5 displays the SI area “Process-Product” (E) being the highest instead of “People-Process-Product” (G) area.

Looking closely at each isolated SI element analysis revealed that the people system was unarticulated in the lessons-learned. There seems to be an assumption that the people-element ‘Developer’ is behind most of it all; thus, the element was not explicitly communicated. With this being said, adding the appropriate people-element (most likely the ‘Developer’) to each combination in the SI area “E” (291 instances) would make area “G” (85+291=376 instances) surpass area “E”’s tally.

The lack thereof people-element is confirmed by the low tally of SI areas “People-Process” (D) and “People-Product” (F). However, using the same concept that the people-system drives the process-system; SI area “Process-Process” (B)’s tally (204 instances) will contribute to area “D” raising its bar (14+204=218 instances). In retrospect, by flipping this concept to having the people-system as the recipients, operators, or users of the product-system; then SI area “F” will consume the tally of the “Product-Product” (C) area (228 instances) also raising its bar (7+228=235 instances). In doing all these suggestions: area “D” increases its tally by 204 (area B’s), area “F” increases its tally by 228 (area C’s) and area “G” increases its tally by 291 (area E’s). Area A should increase, however, the mathematics of the people-element is not as straight forward as the process and product elements. Deliberate “People-People” integrated combination in the context of the lessons-learned used are not articulated due to the lessons-learned authors’ objectives. Figure 6 illustrates a new graph with “normalized” SI areas with an increasing curve denoting the integrated dependency of the three SI-elements.

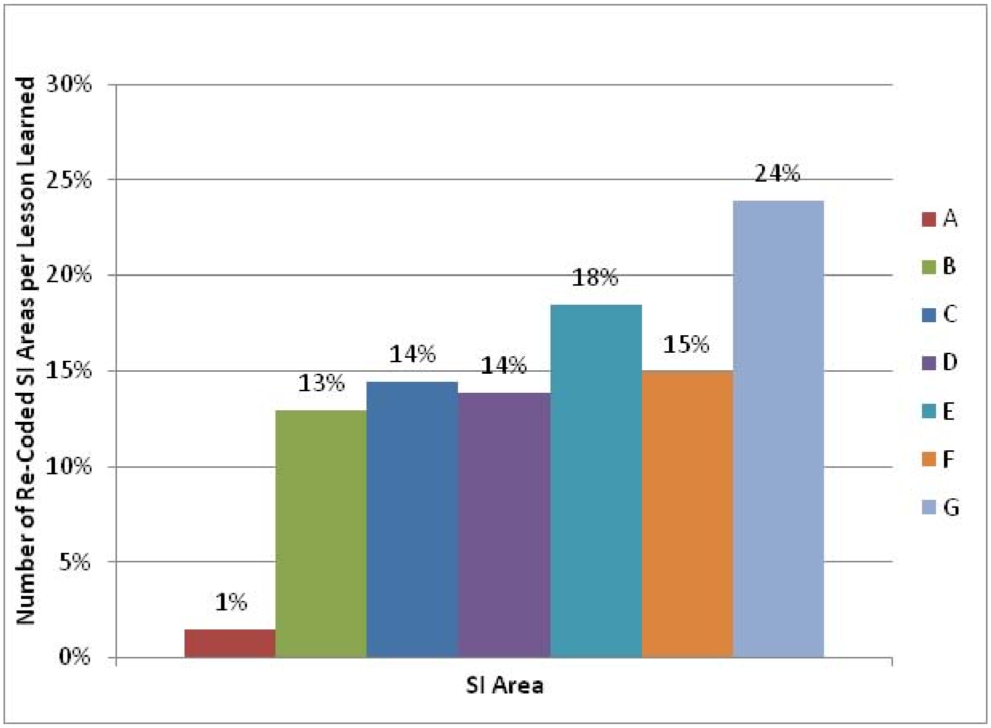

Figure 6. "Normalized" Systems Integration Areas

<u>Multi-Level and Multi-Dimensional</u>

Because of the highest total tally of the SI product-element, SI has to have the same multi-level structure in order to bring together lower product items of a system into higher product-items. This interfacing action is usually when SI is realized and can be measured through the number of items to be interfaced per level. This measurement timing is confirmed by the high tally of the process-element 'Evaluate' which compliments these SI activities ensuring the product-items "work together." Not only does SI need to perform within the levels of each element (represented by SI areas A, B, and C); but also have to synchronize its performance between elements (represented by SI areas D, E, and F). This integration effort between SI-elements interacting, interrelating, and interfacing demonstrate SI's multi-dimensional characteristic.

Traceability Matrix-Model

The ability to trace SI activity throughout the life-cycle of a DoD Space System Acquisition is one of the research objectives of this thesis. The other research objective "What are the characteristics of SI" will be answered in the process of answering the first research objective. One cannot be answered without answering the other.

To understand SI traceability, a simple analogy will be used.

> If your computer fails to boot up when you power it on, you will normally follow a structured sequence of procedures to troubleshoot the problem, fault isolate the problem, and take corrective action. First, the operator must know the high-level functions that are performed in order to power on and boot up the computer. This set of requirements is analogous to the acquisition professional knowing the standard DoD systems acquisition process (e.g., phases, milestones, key decision points, reviews, audits, etc.). The frustrated computer operator must know that a power source (AC or DC) must be applied to the computer. Setting the ON/OFF switch to the ON position is next. After successfully conducting these two tasks, the operator will review progress made up to this juncture and assess whether any adjustments or corrective actions are required before proceeding forward. This process will continue until the computer is up and running, or another course of action is initiated (e.g., call the help desk).

Accordingly, to develop a SI traceability model that can be used to assess program SI; first the SI Areas must be established. This task was performed in Chapter I, Figure 1 and resulted in seven SI Areas. As discussed in Chapter I, SI Areas represent the *people, processes, and products* that *interact, interrelate, or interoperate* in order to successfully manage a DoD systems acquisition program. Each of the SI Areas was further decomposed into SI elements that are involved in the activities prescribed by the TR&A processes. For example; in the People-People SI Area, the people that are interacting may be Acquirers, Developers, Stakeholders, or Operators. The Process-Process SI Area is decomposed in the same manner; for example, definition, design, deployment, or evaluation may be SI elements for this area. All seven SI Areas are decomposed in this manner (see Table 7). Much of this thesis is devoted to the integration of these SI Areas. However, to trace the existence of SI activities occurring in a program and thereby assess the efficacy and adequacy of the program's SI, there must be standards by which SI can be measured.

To form a working standard, SI elements were coded to draw out the conceptual meaning of words and phrases used by the People involved in a DoD systems acquisition program. Coding started with Concept Analyses (10:1) performed on the LL statements. Once the words and concepts that were most relevant and applicable to program management and systems engineering were identified (through concept analysis), they were grouped into predetermined categories. These categories were each assigned a single code word that conveyed the intent of the concepts being expressed by the People using them. Code words were then decomposed in a manner that facilitated the allocation to one or more of the TR&A events (see Table 5).

A matrix consisting of the conventional DoD systems acquisition TR&As, organized by acquisition phases, was constructed with the TR&As listed along the top, and the SI Areas with elements listed along the side. The matrix which is fully populated with "ideal" entries becomes the model or standard, against which a program under review can be measured (see Table 14). Table 8 summarizes the matrix with totals of each SI-Area per TR&A event.

Table 8. Traceability Matrix-Model Summary

		Phase 0: Concept Studies	Alternative System Review (ASR)	Phase A: Concept Development	Integrated Baseline Review (IBR)	System Requirements Review (SRR)	System Design Review (SDR)	Phase B: Preliminary Design	Preliminary Design Review (PDR)	Phase C: Complete Design	Critical Design Review (CDR)	Phase D1: Fabrication & Integration	Production Readiness Review (PRR)	Test Readiness Review (TRR)	System Verification Review (SVR)	Functional Configuration Audit (FCA)	Physical Configuration Audit (PCA)	Phase D2: Fielding & Checkout	Mission Readiness Review (MRR)	Flight Readiness Review (FRR)	Launch Readiness Review (LRR)	Phase D3: Operations & Disposal	Post Flight Review (PFR)
A: People-People																							
25	**TOTAL**		2		4	8	9		9		10		10	10	10	10	10		10	10	10		3
B: Process-Process																							
204	**TOTAL**		11		15	15	17		20		36		38	38	38	38	38		38	38	38		38
C: Product-Product																							
228	**TOTAL**		7		8	17	29		37		37		37	37	37	37	37		37	37	37		37
D: People-Process																							
15	**TOTAL**		4		4	5	7		7		7		7	7	7	7	7		7	7	7		5
E: Process-Product																							
291	**TOTAL**		5		17	23	34		41		60		71	71	71	71	71		71	71	71		71
F: People-Product																							
7	**TOTAL**		3		3	3	4		5		6		6	6	6	6	6		6	6	6		3
G: People-Process-Product																							
85	**TOTAL**		9		9	14	18		35		44		50	50	50	50	50		50	50	50		50

Findings – SI Traceability

The following findings are the results of tracing the SI-areas combination-items (which were extracted from space-systems acquisition lessons-learned) through the lifecycle TR&A transitions.

<u>A: People – People</u>

Analyzing the summary results from the traceability matrix-model as shown in Table 8, this SI-area starts with minimum 'people-people' interactions in the early phases (i.e. Concept Studies and Development phases) and gradually increases as the program lifecycle transitions to the design phase (i.e. Preliminary Design phase). From here, the SI area "A" interactions transitions constantly through Operations & Disposal phase where these interactions drop to the minimum. However, this drop should not be an indication that there are less 'people-people' interactions in this phase; rather it confirms that this SI area's combination-items which were extracted from the LLs are focused on the design and development phases. Perhaps, these phases represent the program's critical lifecycle stages where SI interventions and practices are mostly in need. Looking closely to the details of these combination-items, the people type 'Developer' interacting with another 'Developer' and other people-types (i.e. 'Operator' and 'Supplier') supports the concentration of TR&A events that traces from System Design Review (SDR) through Launch Readiness Review (LRR). In addition, the drop during the Post Flight Review (PFR) indicates the people-type 'Developer' having the "main-character" role does not transition throughout the program's lifecycle – a disconnect?

B: Process – Process

There is a similar increasing trend to the traceability of the 'process-process' interrelationships but shoots up during the Critical Design Review (CDR), reaches its peak at Production Readiness Review (PRR) and remains constant through the Post Flight Review (PFR). The CDR event emphasizes the start of the 'Evaluate' process (this process-element ranked the highest); whereas the 'Define-Design' processes starts the traceability from the beginning TR&A, that is Alternative System Review (ASR). Hereinafter, the "chain" of processes is iterative due to changes and modifications. This practice indicates the importance of tracing SI to ensure a seamless flow from inputs to outputs of all processes. Again, the same segment of phase transitions show the need of focused SI practices.

C: Product – Product

The value of SI within 'product-product' interfacing-activities starts with SDR and traces constantly throughout the program's lifecycle. The traceability trend is very similar to the 'process-process' SI area due to the fact that these processes bring about the product system. The early phase TR&A events trace SI activities that are mostly involved with conceptual, high-level product-type 'Requirement' (the second highest product element). Understandably, these product-types drive the "fanning" decomposition of the product-system where SI practices are critical to the program's success. SI objectives should ensure interconnectivity flows down and feeds-back up the infrastructure of the product system. The highest ranking product element, 'Hardware', as extracted from the LLs confirms this 'Requirement'.

D: People – Process

This SI-area is the first of crossing SI-elements and tracing their "working together" practices. The trend of the traceability shows a "bell-curve" starting and ending with minimum 'people-process' integration. This trend confirms that the people-type 'Developer' (highest ranking) heavily engages during the 'Define-Design-Implement-Evaluate-Deploy' processes; whereas the other people-types, like the 'Acquirer' and 'Operator', engage themselves at the beginning and end of the lifecycle and provide oversight to the 'Developer's' activities. Oversight by these type of people is necessary because the products brought about by the 'Developer's' processes are ultimately for their utilization.

E: Process – Product

This next SI-area continues the SI traceability indicating the height of SI's value during Fabrication & Integration Phase. The top ranking process-element 'Evaluate' and product-element 'Hardware' confirms the high frequency tally of this SI-area's combination-item, 'Evaluate-Hardware'. This SI-area's traceability trend is the same as the 'process-process' integration which demonstrates that the product system is the outcome of the process system.

F: People – Product

The 'people-product' SI-area shows a similar "bell-curve" as the 'people-process' SI-area. This similarity shows that the practice of SI begins and ends with the people system. Needless to say, the process and product systems are quite useless without the people system steering the flow to success.

G: People – Process – Product

This SI-area culminates the actions of "working together" between and among these SI-elements which are shown individuality in SI-areas "A" to "F". This dependency and linking phenomena confirms that the practice of SI is "People-Driven", "Process-Centered", and "Product-Oriented". The traceability trend of this SI-area will be similar to SI-areas "B", "C", and "E" where their activities start increasing to a constant value. Perhaps, a saturation point controlled by the process-system's definition of the product-system.

Applying Traceability Matrix-Model with GPS Case Study

The same traceability approach in establishing the matrix-model as previously described is used in assessing a DoD space-system program. The DoD space system program selected for this assessment is GPS, based on the Case Study performed by the Air Force Institute of Technology (AFIT). Results from this assessment (see Table 15) have no relevance to the program's success (or lack of), or the quality of SI being performed. The assessment only demonstrates the ability to trace SI using TR&A transition criteria as evidence to answer investigative question number two.

This program has been chosen for its program information availability, recent implementation and comprehensive post-milestone program analyses. Information for the program under review is required to determine the TR&As accomplished during an acquisition phase of interest. Recent program implementation is important to align case study TR&A events with standard DoD Systems Acquisition TR&A events used in this thesis. Post-milestone program analyses are required to validate the SI traceability resulting from the assessments performed on the programs that were reviewed.

The Traceability Matrix-Model will be applied to the TR&A events derived from the GPS case study. Words and phrases used in the GPS case study to indicate that a TR&A event had been accomplished were coded to ensure a standard process for gathering the elements used in the traceability matrix. Using the coded words, the GPS case study was searched for these word and their associated TR&A events. These events were then tallied according to the SI Areas indicated by the coded words and the tallies inserted into the traceability matrix. The subsequent GPS case study traceability matrix was then compared to the Standard traceability matrix. If the two matrices were closely aligned in proportionality, then it was deduced that the GPS program represented by the case study was employing SI sufficiently. If, however, a comparison of the two matrices indicated that there was a significant difference, then it could be deduced that the GPS program was not employing SI sufficiently.

Table 9 summarizes the traceability matrix for the GPS Case Study; whereas Table 15 populates the traceability matrix model with the program data presented by the case study. Keep in mind that this GPS case study's objectives focused on Systems Engineering; thus may not be closely relevant to the SI and TR&A concepts that can be traced using this thesis' traceability method.

Table 9. GPS Traceability Matrix Summary

		Phase 0: Concept Studies	Alternative System Review (ASR)	Phase A: Concept Development	Integrated Baseline Review (IBR)	System Requirements Review (SRR)	System Design Review (SDR)	Phase B: Preliminary Design	Preliminary Design Review (PDR)	Phase C: Complete Design	Critical Design Review (CDR)	Phase D1: Fabrication & Integration	Production Readiness Review (PRR)	Test Readiness Review (TRR)	System Verification Review (SVR)	Functional Configuration Audit (FCA)	Physical Configuration Audit (PCA)	Phase D2: Fielding & Checkout	Mission Readiness Review (MRR)	Flight Readiness Review (FRR)	Launch Readiness Review (LRR)	Phase D3: Operations & Disposal	Post Flight Review (PFR)
A: People-People																							
25	**TOTAL**		0		5	2	4		2		4		5	0	0	5	0		0	0	0		0
B: Process-Process																							
204	**TOTAL**		0		11	7	6		9		7		7	0	0	4	3		0	0	0		0
C: Product-Product																							
228	**TOTAL**		0		8	1	7		0		11		9	0	0	0	6		0	0	0		0
D: People-Process																							
15	**TOTAL**		0		3	0	3		0		2		5	0	0	2	0		0	0	0		0
E: Process-Product																							
291	**TOTAL**		0		0	0	1		0		0		0	0	0	0	0		0	0	0		0
F: People-Product																							
7	**TOTAL**		0		0	2	0		1		0		0	0	0	2	2		0	0	0		0
G: People-Process-Product																							
85	**TOTAL**		0		11	4	0		7		0		8	0	0	1	0		0	0	0		0

Findings – GPS Program's SI Traceability

Comparing the tallies from the GPS Case Study SI Traceability matrix and those from the Standard SI Traceability matrix (see Table 10) it is clear that there is no direct linear relationship between the two. This outcome was expected and is explained by the variability in individual space-systems acquisition programs. The Defense Acquisition System recognizes the variability in individual DoD acquisition programs and accommodates this variability. Therefore, while it is possible to use the SI traceability methodology (including the matrix) to trace the application of SI occurring in a space system acquisition program, it is not possible to quantitatively measure SI adequacy.

The GPS Case Study SI Traceability example clearly demonstrates that SI can be traced throughout the acquisition lifecycle of a space-system acquisition program. This example similarly demonstrates that the tally of TR&A events that occurred in the GPS program cannot be used to assess SI adequacy of the GPS program. One reason for the variance between the Standard SI Traceability Matrix and GPS Case Study SI Traceability Matrix TR&A tallies is that the GPS Case Study consisted of only three phases, Phase I – Concept/Validation, Phase II – Full-Scale Engineering Development and Phase III – Production. (20:27) Furthermore, the GPS program that the case study examined started in 1973 and achieved Full Operational Capability (FOC) in 1984; which means that the DoD Systems Acquisition process (i.e., TR&As, phases, etc.) was different from that used for the Standard SI Traceability matrix. (20:27) The GPS Case Study SI Traceability example does, however, provide a tool which can be used to assess the phasing and types of TR&As being conducted to support an acquisition program. It will remain the responsibility of the Program Manager and the Chief Engineer to interpret the results of this tool. Professional judgment and knowledge must be brought to bear on the SI Traceability Matrix results. The PM and CE can then project the interpreted results onto the actuals of a given acquisition program and derive a satisfactory interpretation of the results and plan a course of action that will add value to their program.

Table 10. GPS Case Study vs Standard SI Traceability

Phase 0: Concept Studies	Alternative System Review (ASR)	Phase A: Concept Development	Integrated Baseline Review (IBR)	System Requirements Review (SRR)	System Design Review (SDR)	Phase B: Preliminary Design	Preliminary Design Review (PDR)	Phase C: Complete Design	Critical Design Review (CDR)	Phase D1: Fabrication & Integration	Production Readiness Review (PRR)	Test Readiness Review (TRR)	System Verification Review (SVR)	Functional Configuration Audit (FCA)	Physical Configuration Audit (PCA)
A: People-People															
	0/**2**		5/**4**	2/**8**	4/**9**		2/**9**		4/**10**		5/**10**	0/**10**	0/**10**	5/**10**	0/**10**
B: Process-Process															
	0/**11**		11/**15**	7/**15**	6/**17**		9/**20**		7/**36**		7/**38**	0/**38**	0/**38**	4/**38**	3/**38**
C: Product-Product															
	0/**7**		8/**8**	1/**17**	7/**29**		0/**37**		11/**37**		9/**37**	0/**37**	0/**37**	0/**37**	6/**37**
D: People-Process															
	0/**4**		3/**4**	0/**5**	3/**7**		0/**7**		2/**7**		5/**7**	0/**7**	0/**7**	2/**7**	0/**7**
E: Process-Product															
	0/**5**		0/**17**	0/**23**	1/**34**		0/**41**		0/**60**		0/**71**	0/**71**	0/**71**	0/**71**	0/**71**
F: People-Product															
	0/**3**		0/**3**	2/**3**	0/**4**		1/**5**		0/**6**		0/**6**	0/**6**	0/**6**	2/**6**	2/**6**
G: People-Process-Product															
	0/**9**		11/**9**	4/**14**	0/**18**		7/**35**		0/**44**		8/**50**	0/**50**	0/**50**	1/**50**	0/**50**

Note: TR&As end after Phase D1 due to the GPS program strategy
Legend: Tallies are given by – GPS/**Standard** (where Standard tally in **bold**)

Findings – Measuring Integration

Although measuring integration is not the primary focus of the research, the hope is that the course of this research will eventually lead to some other measurements of SI. The following measurements were the most commonly mentioned indicators relevant to SI which this research corroborates with its results from applying the proposed models to the Space System Acquisition framework.

Traceability

Since SI is multi-dimensional and hierarchical (i.e. multi-level) in nature, the capability to trace its performance can be measured. Traditionally, SI's traceability is measured within the "Product-Product" area through diligent analyses and control of its element-type 'Requirement' (specifically, interface control requirements). The product-element type 'Requirement' is enabled by the process-element type 'Definition'. This link in itself is an SI traceability factor. This thesis' research also includes tracing SI across the Space System Acquisition framework using the TR&A transitions. Traceability is also dependent on the process of architecting the three SI-elements. Their decomposition is represented by breakdown-structures, commonly known as Work Breakdown Structures (WBSs).

Complexity

This measurement is another effect of SI's multi-dimensional and multi-level character. SI complexity is viewed and allocated differently by each of the three SI-elements. Typically, SI complexity is assigned the difficulty to successfully achieve the action of "working or bringing together" system elements. The most common viewpoint is confined to the product-element which is demonstrated by its high frequency tally. SI-Product complexity is defined as the physical number of objects, units, or components to be integrated, that is, internal or external to a location-point in the hierarchy. For the process-element, SI complexity is measured as the number of tasks, activities, and timing to be integrated. The SI-Process elements extracted from the lessons-learned and their interrelationship between and among the other SI-elements bring about the amount of effort to consistently control the similarities and differences of each process. Whereas, the SI complexity of the people-element applies to the number of individuals, specialties, and amount of authority involved. All encompasses the participation of these actions of integrating the development of the overall system solution.

Connectivity

Making various pieces of often disparate equipment work together defines the measurement of connectivity. This measurement relates to the SI complexity of the product-elements addressing the lower-level physical (i.e. form and fit) characteristics of the interfaces (e.g. the number of input and output ports). Successful connectivity relies heavily on the process of requirements definition, specifically between interfaces represented by interface control documents (ICDs).

Interoperability

Interoperability applies to the functionality (as oppose to physicality) of one SI-element working together with other elements. During this action of integration, the SI-element's capabilities are exploited to deliver an overall system solution that minimizes operations of stovepipe systems. Generally, this measurement entails the working together of the data structures of specific functions of the physical product-elements. Successful interoperability depends on the seamless interrelationships of the processes that develop the products resulting in a suitable and effective system. This integrated system is expected to interoperate (minimum operator intervention) with other systems used during mission employment.

Compatibility

Compatibility has an inherent human-element in it. Thus, this measurement focuses on the people-element of SI complexity. The SI point of view depends on the acquisition phase the system solution is in expanding or leveraging the people-element's skills and capabilities in accomplishing the difficult tasks of the process-elements. This measurement applies to the consistent control of the similarities and differences of the specialties involve in achieving the objectives of each space-system acquisition phase from concept development to operations and support.

Implications and Interpretation

Through "life integrating" analogies, this section describes the implications and interpretations on SI as a whole. It is one way of emphasizing the people element in the SI construct. These analogies are based on the analysis of the lessons learned, the review of the subject matter's literature, and the researcher's experiences on actions of "working together."

'SI's Taste of Mango'

Describing the taste of mango to someone who has never tasted one is like describing SI – very challenging. More often than not, description is transferred to trying the mango. This is because the taste of mango cannot be directly observed and similarly the construct of SI is underlying to the fundamentals of doing anything within the system being integrated. This tendency to expect a physical or visual description of SI is demonstrated by the high tally of the product-element extracted from lessons-learned which were humanly written. Whereas, the high tally of the process-element implies "experiencing" SI like trying the mango.

<u>'SI Takes Two to Tango'</u>

Like watching two people beautifully dance the Tango, it takes at least two element-types to successfully demonstrate the action of working together – to link one level with another level, to perform one function with another function, to build one component with another component, or to design a requirement with tools and materials into things. As listed in the traceability matrix-model (Appendix C – Table 14), the combinations of SI-elements in the SI areas (A, B, C, D, E, F and G) confirm this phenomenon. Additionally, Figure 5 Figure 5shows SI areas B, C, and E as the toppers with E as the forerunner. The combinations in the "Process-Product" (E) area reveal that processes with different objectives have to work in the same, synchronized patterns to produce integrated sound quality products at all levels of the hierarchy. Also, the people behind the scenes with different experiences have to "dance" their way in processing products in the same program.

<u>'SI's World Trade Center'</u>

As tragic as the toppling down of the World Trade Center, SI's hierarchical nature topples down when one element is threatened. SI structure is realized by its product-element. The raw product-type descriptions in the lessons-learned traced the root-causes down to the smallest unit possible. This analysis would not have been as detailed if not for the design descriptions. Thus, the process system must have the same levels of abstraction as the product system complimenting each other. These actions are driven by a similar organizational hierarchy in order to control and coordinate the transitions.

‘SI of All Trades, Master of None’

Inherently, SI requires cursory understanding of the attributes, properties, characteristics, and performance of the three sets of system-elements. SI does not need to master any of its elements’ behavior in order to successfully execute; rather this mastery would influence the balance of the three elements. Therefore, in this case, ‘master of none’ is a positive concept and significantly desirable. However, skillful mastery in SI requires the know-how to deal between and among its elements. SI needs to know what and how to discern each element’s behavior for trade-offs. This skill just emphasizes SI is event-driven.

‘Bottoms-Up, SI’

Traditionally, SI is realized with a bottom-up approach starting at its lowest-level of abstraction and ending with the overall system solution. The amount of SI efforts in the lower levels is at its maximum tapering down as the work rolls-up to the top.

‘SI Divides and Conquers’

Decomposing and aggregating the SI-elements together demonstrate SI’s need for a top-down approach to successfully ‘conquer’ its objectives.

‘SI’s Power of Suggestion’

SI’s power of suggestion makes its character obvious to most people. As obvious as it may seem, SI suggests underlying requirements and repetitive cycles to master its execution.

‘No SI-Element is an Island’

Each SI-element cannot work alone to achieve an overall system solution.

Despite their own objectives, SI takes melding the objectives of all three elements to deliver an integrated system solution for mission utilization. From the data analysis of this thesis, SI is characterized as "people-driven, process-centered, and product-oriented" brought together to form an integrated "people-process-product" entity.

<u>'SI's Twin Brother SE'</u>

Despite SI's operative uniqueness, SE approaches are supportive of its successful application. For instance, system engineering management supports program management's ultimate responsibility for the products of SE, for managing risk, and controlling the configuration of the products that make up the system. It can be useful to consider SE as a cross-product process and the SE organization as a cross-product staff function serving program management. The technical and engineering program aspects should be addressed as an integrated part of the SE process. Successful integration of engineering and technical elements in acquisition programs is dependent upon substantive and proactive organizational processes. Systems Engineering must be an integrated system capable of providing and sustaining the people, products, and processes necessary for the effective and efficient execution of the program objectives. To achieve the stated objectives of systems engineering there must be a process for planning, directing, monitoring, and controlling all the engineering on a program. This process is what we interpret as Systems Integration.

Investigative Questions Answered

SI traceability performed using the GPS case study revealed that, given a moderate amount of program management artifacts, SI can be traced throughout the lifecycle of a standard DoD system acquisition. By using the SI Traceability Matrix-Model, a structured, linear, and repeatable methodology can be applied to any DoD system acquisition program in a manner that will reveal information about the program under review. This information, when subjected to the Traceability Methodology developed in this thesis, will enable a knowledgeable DoD Systems Acquisition Professional, to formulate an assessment of a program's SI activities and characteristics as revealed by programmatic lessons learned. Once this assessment is rendered, the appropriateness and adequacy of the program SI must be formulated by the individual(s) performing the SI analysis.

In the thesis example of SI traceability analysis (i.e. GPS case study), it was clearly demonstrated that SI can be traced throughout the lifecycle of a DoD space-system acquisition using standard DoD TR&A events. Using this same SI traceability methodology, but with a greater degree of analysis, the level of program SI activity, efficacy of program SI, and performance of SI relative to program objectives, can also be derived.

Summary

By rigorously employing the methodology described in this thesis, it has been demonstrated that SI in a DoD Space-System Acquisition program can be traced and its performance can be measured.

Top-level SI-element complexity (e.g., the number of sub-elements associated with the SI Area) will render the SI traceability methodology seemingly incomprehensible. However, by collecting, characterizing, grouping, and coding the words and phrases that link the applicable programmatic, technical and system concepts to the standard DoD Systems Acquisition Technical Reviews and Audits, it is possible to devolve these complexities into manageable parameters that can be reasonably managed. By extracting the SI-elements from lessons-learned and problem reports, their interrelationships with the other SI-elements can portray the amount of effort needed to consistently implement each process in the Traceability Methodology. Complexity of other SI elements may depend upon such factors as the number of stakeholders, individual specialties, and amount of decision authority involved in the TR&A process. The analytical process of projecting three individual SI Area elements onto one TR&A events becomes exceedingly complex and subjective when too many contributing factors are involved. As a consequence of the additive complexity factors that are inadvertently introduced into the Traceability Model, the authors of this thesis chose to take a simple and relaxed approach to data collection, reduction and interpretation.

When data collection, reduction and analysis resulted in the high tally of the SI Area elements associated with TR&A events, a quantitative measure of SI activity for a DoD Systems Acquisition program that is under review could be compared to a standard and a measure of the program SI performance could then be assessed. The characteristics of SI are revealed by lessons learned from various space-system programs and proved to be relatively consistent between programs and systems.

A significant realization that each SI-element could not function alone to achieve an overall system solution was revealed during the course of this thesis. Despite the individual objectives of each SI Area, SI requires the melding the objectives of all three areas, their intersections and unions, to deliver an integrated system solution for program success.

V. Conclusions and Recommendations

This chapter summarizes the research in characterizing SI and tracing its value along the TR&A timing in the Space System Acquisition framework. Descriptions include the thesis' contribution to the theory and practice of SI, its research limitations, recommendations for action, and suggestions for future research. The chapter ends this composition with conclusions that can be drawn from its research.

Contributions of Research

This thesis explored the subject of Systems Integration as applied to the Space-Systems Acquisition process. Despite the unsupported literature review on this subject area, this thesis created two new models. The first model proposed a three SI-element construct of seven SI-areas that served as the foundation of the second model. The second one was created to trace SI within the Space-System Acquisition framework. It served as a measurement tool for this research and was partially validated with one case study.

Characterizing SI through lessons-learned and tracing these characteristics within a framework cracked opened a little more the door of integration knowledge. Although not part of the plan, the research also brought about some ways to measure SI. The following paragraphs briefly discuss the more important contributions to the theory and practice of SI in the Space Systems Acquisition framework.

Prior to this research, there were little specific and disparate information on SI. The findings show that SI is vaguely defined and difficult to articulate although desirable for a number of reasons.

The data collection and analysis strongly supported the view that SI is multi-level and multi-dimensional with three elements combined into seven areas. The evidence supports the definition of SI as "the work of many people from different functional disciplines, working on different system component products, in different process steps over time" (21:42).

Several ideas were found that might allow further discoveries of measuring SI leading to new functions for program estimation.

Initial efforts have been taken towards developing a tool that assigns value to SI. The ability to trace SI in the acquisition framework permits the assignment of values to SI-area. By establishing the links between and among these SI-elements may provide the ability to establish cost benefits of SI.

Finally, this research's confinement to the context of space-system infrastructure should not deter the application of the many same concepts to other types of infrastructure. Ultimately, the idea that SI construct can be universally applied will likely be confirmed.

Limitations of Research

As far as the authors' efforts to finding similar approaches, this research is the first to characterize and trace SI in the Space Systems Acquisition framework. However, the reader is informed of the several limitations of this research as describe in the following:

- Being the first in this type of study requires additional research to confirm the results.

- Additional case studies are needed to fully validate the research models.
- Need to validate concept coding to confirm the research's findings.
- This research is not as strongly grounded as usual due to lack of specific literature.
- Lack of universal SI measurements limited stronger relationships between and among the people, process and product elements.

Recommendations for Future Research

Additional exploration is needed for this course of research. There are a variety of opportunities to widen the door of SI. Some specific ideas are:

(1) Create a portfolio of measurements to estimate the value of SI in space-system acquisition programs.

(2) Utilize this research's models to assure SI is employed within the program's objectives and scope.

(3) Find the downsides of SI to establish guidelines for SoS program decisions.

(4) Investigate differences of perceptions about SI from both the government and industry.

Conclusions

The subject area of Systems Integration is ample and intricate. SI appears to be hierarchical in nature and multi-dimensional consisting of three elements combined into seven areas. Evidence suggests that SI is being pursued by many space-system programs based on its obvious definition of “working together” and without any means to ascertain its value.

This research has discovered some of the theoretical issues of SI. Although not encompassing, this course of research encourages SI practitioners to look at it with more actual ideas. This additional knowledge provides practitioners and researchers alike with greater leverage to make more intelligent and informed decisions.

Despite the efforts of characterizing SI and tracing its value within the Space Systems Acquisition framework, it still remains to be seen the articulation and standardization of this phenomenon. Logically, this lack of common language should stifle SI’s great potential value to succeed, however, practitioner’s heavy reliance on their intuition and judgment for SI decisions seem to have served remarkably well.

Appendix A: Collection of the Lesson-Learned Statements

1 Honeycomb Structures Should be Vented to Reduce Delamination Risk

- *Honeycomb structures* for *space systems* should be *design*ed vented whenever possible.
  C Product–Product: honeycomb structure – space system
  E Product–Process: honeycomb structure – design
- The vast majority of *spacecraft or launch vehicles* using unvented *honeycomb structures*, and these have not failed in *space operations*.
  C Product–Product: honeycomb structure – spacecraft, launch vehicle
  E Product–Process: honeycomb structure – space operations
- If an unvented *design* cannot be avoided (e.g., to avoid contamination), it is necessary to adopt extensive *development*, *verification*, and *quality assurance*, including proof *tests* under applicable *temperature* and *vacuum* conditions.
  B Process–Process: design – development, verification, QA, test
  E Process–Product: design – temperature, vacuum (requirements)
- Perforating *honeycomb cells* relieves pressure during *launch*.
  E Product–Process: honeycomb cells – launch

2 Perform Independent Mass Property, Stability Control, and Structural Load Analyses on Spacecraft and Launch Vehicles

- *Engineer*'s inaccurate *modeling* on *mass property*, *stability control*, and *structural loads* continue to threaten *mission performance*.
  G People - Process - Product: engineer – modeling – mass property, stability control, structural loads
  B Process–Process: modeling – mission
- Many *programs* require an independent *analysis* to ensure correct *modeling*.
  D People–Process: program – analysis
  B Process–Process: analysis – modeling
- Independent *analyses* also help *validate* operational *procedures* and *support* flight anomaly *resolution*.
  B Process–Process: analyses – validate – support – resolution
  E Process–Product: validate – procedures
- Independent *analysis* is often necessary to overcome each *organization* having little insight into each *other's* analytical process.
  D Process–People: analysis – organization
  A People–People: organization – other
- Integrating *space vehicle (SV)* to *launch vehicle (LV)* involves complex *modeling*.
  C Product–Product: space vehicle – launch vehicle
  E Product–Process: space vehicle, launch vehicle – modeling
- *Costing* problems can easily arise without a clear settling of *organization*al responsibility, especially with today's emphasis on *proprietary data* protection.
  G People - Process - Product: organization – cost – proprietary data

3 Rigorously Manage and Test Software, Including the Database

- Multiple deficiencies in the *software development*, *test*ing, and *quality assurance* (QA) processes allowed a single–Point failure escape during *satellite operations*.
  E Product–Process: software – development, testing, QA
  B Process–Process: development, testing, QA – satellite operations
- *One* must *test* actual flight *hardware and software*.
  G People - Process - Product: one – test – hardware and software
  E Process–Product: test – hardware and software
  C Product–Product: hardware – software
- Due to a lack of *overall* software *ownership*, *independent validation* and *verification* was not done on the as–flown *constant*.
  G People - Process - Product: Ownership – verification and validation – constant
  A People–People: overall – independent
- The integrity of software *databases* is no less critical than the *source codes*.
  C Product–Product: databases – source code
- The *space business* is extremely complex and *human error* cannot be completely eliminated.
  A People–People: space business – human error
- The *system* must be *design*ed robust enough to catch the inevitable faults.
  E Product–Process: system – design
- The wrongly placed *decimal point* caused the middle line *display* to become flat.
  C Product–Product: decimal point – display
- *Satellite operator* failed to read the flagged anomaly on *display* during *launch*.
  G People - Process - Product: Satellite Operator – Launch – display

4 Document Engineering Requirements As Clearly As Possible

- *Engineers* must clearly articulate their intentions and determine how the *requirements* should be interpreted or could be misconstrued.
  G People - Process - Product: Engineers – determine – requirements
- *Builders* making seemingly minor (Category II) *changes* can misconstrue their interpretation of unarticulated *requirements*.
  G People - Process - Product: builders – changes – requirements
- *QA* was led to pass joints with low brazing coverage due to missing "per linear inch" phrase in the *requirements*.
  E Process–Product: QA – requirement
- The work *instruction* for assembling joints stated that the wrapping should be applied "within 0.5 inches of the mounting bracket flange" (instead of saying, e.g., no closer than 0.5 inches) caused to breach combustion *chamber* during *flight operations*.
  C Product–Product: instruction – chamber
  E Product–Process: chamber – flight operations
- The *technicians*, not knowing that the parts were to unfasten, *built* by taping the *joints* as closely to the *flange* as possible, making separation impossible (breached) during *flight operations*.
  G People - Process - Product: Technicians – built – joints and flange
  C Product–Product: joints – flange
  E Product–Process: joints and flange – flight operations

5 Avoid Pure Tin Plating

- Prohibit *design* of pure tin plating in both *flight hardware* and *ground equipment*.
  E Process–Product: design – flight hardware and ground equipment
  C Product–Product: flight hardware – ground equipment
- *Buyers* ensure *prime contractors flow down* unambiguous plating *requirements* and perform appropriate receiving *inspections*.
  A People–People: buyers – prime contractors
  G People - Process - Product: prime contractors – flow down – requirements
  B Process–Process: flow down – inspections
- Requirement *planners* appropriately *store & handle* (i.e. purge) prohibited tin materials (paying particular attention to the *"commercial parts"*) from *project stores* and standard *catalog items*.
  G People - Process - Product: Planners – storage & handling – commercial parts
  C Product–Product: commercial parts – project stores, catalog items
- *PM review subcontractor* designs and part *specifications* to confirm *safety* of *parts*.
  A People–People: PM – subcontractor
  G People - Process - Product: PM – review – specifications
  E Product–Process: parts – safety
- Apply conformal *coatings* on all exposed conducting *surfaces* wherever possible to inhibit shorts and vacuum arcing during *test*.
  E Process–Product: coatings – surfaces
  B Process–Process: coatings – test

6 Following a Major Repair, Watch Out for Secondary Damage

- Ad–hoc *repair* processes tend to be much less defined and qualified than regular *manufacturing* operations.
  B Process–Process: repair – manufacturing
- Material Review Board (*MRB*) should *review* possible secondary damage from *repair*ed *rocket* and should be added to the *readiness review* process.
  G People - Process - Product: MRB – repair review – rocket
  B Process–Process: repair review – readiness review
- *Repair* patching of deep cut on *rocket* segment allowed flame to burn through the *case*.
  E Process–Product: Repair – rocket
  C Product–Product: rocket – case

7 Perform High–Fidelity System Validation Tests for Pyrotechnics

- *Pyros* (explosive devices) by themselves are very reliable, but the adjacent *systems* must be *design*ed to withstand (i.e. *test*) the mechanical or electrical shocks generated by the pyros.
  C Product–Product: pyros – systems
  B Process–Process: design – test
  E Product–Process: pyros – design, test
- *Tests* should *simulate flight configuration* and functional *performance*.
  B Process–Process: test – simulate – configuration
  E Process–Product: test – simulate – configuration – flight performance
- *Post–test* examinations of *qualification or acceptance* specimens should look for signs of inferred *margin* or incipient *failure modes*.

B Process–Process: post–test – qualification or acceptance
E Process–Product: post–test – margins and failure modes

8 Solar Arrays Must Withstand Extreme Environments

- *Solar array parts* should be carefully *design*ed from being damaged by the hostile *space environment*.
  C Product–Product: Solar array – parts
  E Process–Product: design – solar array
  B Process–Process: design – space environment
- *Satellites* must be robustly *designed* to withstand the extremes of *space weather* as well as other space hazards.
  B Process–Process: design – space weather
  E Process–Product: design – satellite
- Insufficient *stress relief* and *insulation* caused abrasion of *wiring harness*.
  C Product–Product: stress relief and insulation – wiring harness

9 Excessive Handling Can Destroy Solid Lubricant

- *Operation, testing, or storage* of mechanisms under non–vacuum *conditions* must be performed with caution when MoS_2 dry *lubricant* is involved.
  B Process–Process: operation – testing – storage
  E Process–Product: operation, testing or storage – lubricant, conditions
- Follow *Aerospace*'s *handling and storage* guidelines to safeguard *lubricants*.
  G People - Process - Product: Aerospace – handling and storage – lubricants
- High gain *antenna* unfurls like an umbrella due to excessive friction during *deployment* developed between the *pin* and the *socket* due to loss of *lubricant*.
  C Product–Product: antenna – pin – socket – lubricant
  E Product–Process: antenna – deployment
- The *motor* could not overcome the friction due to lack of *lubricant* and stalled during *deployment* causing the *antenna* to not open.
  C Product–Product: motor – lubricant – antenna
  E Process–Product: deployment – antenna

10 Design Satellites to Withstand Space Weather, Regardless of Solar Cycles

- *Spacecraft* must be *design*ed to withstand worst–case *space environments* as a matter of course.
  E Product–Process: spacecraft – design
  B Process–Process: design – space environment
- *Satellites* should be hardened against electro–static discharge (*ESD*), using well–established *design* guide–lines on *structure, materials, shielding, cable interfaces,* and *circuits*.
  E Process–Product: design – satellites, ESD
  C Product–Product: structure – materials – shielding – cable – circuits

11 Carefully Evaluate Satellite–Launcher Interface

- *Cables* and *connectors* must be *design*ed to withstand vibration–induced *stress*es between *systems* during *test* and *operations*.
  C Product–Product: cables – connectors – systems
  E Product–Process: cables, connectors, system, stress – design, test
  B Process–Process: design – test – operations
- *Margins* must be reserved both in dynamic input *estimation* and in *design*.
  E Product–Process: margins – estimation, design
  B Process–Process: estimation – design
- The interfaces among different organizations, particularly between the *spacecraft side* and the *launcher side*, frequently lead to problems. Independent *analysis* is advised to overcome *organization*al barriers.
  A People–People: spacecraft side – launcher side
  D Process–People: analysis – organization
- The *booster*, while being carried by the launching *airplane*, vibrated at 40–50 Hz. In several previous *flights*, shaking went beyond the level spelled out in the Interface Control Document (*ICD*). As a result, the rocket *contractor reduce*d the airplane's *speed* to minimize this problem. Still, *vibration* in this flight was double the *specification*.
  C Product–Product: Booster – airplane; vibration – specification
  E Process–Product: flight – ICD
  G People - Process - Product: contractor – reduce – speed
- The *satellite* exhibited a structural *resonance* at 40 Hz. During factory *test*, this resonance amplified an acceleration input six–fold.
  C Product–Product: satellite – resonance
  E Process–Product: test – resonance
- The satellite *contractor* conducted the vibration acceptance *test* at a lower level than the *ICD* specification. A defect in the

electronics or *harness* probably went undetected in the *test*, but propagated under a combination of excessive in–flight *vibration* and *resonance* to cause the failure.
G People - Process - Product: contractor – test – ICD
E Product–Process: electronics – test
C Product–Product: electronics – harness – vibration and resonance

- *Vibration*al forces, expressed as power spectral density (PSD) in log scale imparted on the *spacecraft* by the *carrier* airplane, and as *satellite*'s response toward an even level of excitation. Spacecraft resonated at the frequency where above–spec shaking took place during *flight*.
C Product–Product: vibrations – spacecraft – carrier
E Process–Product: flight – frequency, satellite, carrier
- Both the *launcher* and the *satellite* prime *contractors* recognized the *vibration* issue and proposed to conduct a coupled–loads *analysis*. It was not performed because the *program office*, which served as the overall *systems integrator*, lacked *funding*.
A People–People: launcher contractor – satellite contractor – program office – systems integrator
F People–Product: contractors – vibration
E Process–Product: analysis – vibration
G People - Process - Product: systems integrator – funding, analysis – vibration
C Product–Product: satellite – launcher

12 One Requirement, One Statement

- Do not lump several *requirements* together—write them out separately so that each can be *track*ed individually. Negative statements (e.g., “Sampling shall *not* begin until…”) may cause misunderstanding and should be avoided.
C Product–Product: requirement – requirement
E Process–Product: track – requirement
- *Systems engineers* must take ownership of *requirements* and partition (i.e. *flow down*) them to the appropriate *subsystem*. Whether or not a requirement is the *software*’s responsibility, for example, should not be left to the discretion of the *software team*.
G People - Process - Product: system engineers – flow down – requirements
C Product–Product: requirement – subsystem – software
A People–People: system engineers – software team
- *Systems engineer*ing must ensure thorough end–to–end failure mode *testing*.
G People - Process - Product: systems engineer – testing – failure mode
- The software *review* process should emphasize *logic flow*. *Test*s should exercise every *requirement* to see if there are conditions that could cause the *software* to fail.
B Process–Process: review – logic flow
E Process–Product: test – requirement; review – software
C Product–Product: requirement – software
- *Test* planning needs to consider *requirements* for transients or spurious signals.
E Process–Product: test – requirement
- When important *tests* are aborted or are known to be flawed, they must be *rerun* after the errors are *fix*ed. Repeat the *test* if any *software or hardware* involved are changed.
B Process–Process: test – fix – rerun
E Process–Product: test – software or hardware
C Product–Product: software – hardware

13 Flexible Solar Arrays Are Susceptible to Thermally Induced Vibrations

- Flexible *solar arrays* and *support*ing *equipment* are sensitive to *thermal environment*.
C Product–Product: solar arrays, equipment – thermal environment
E Process–Product: support – solar arrays
- Thorough thermo–mechanical *analyses* of the *solar arrays*, particularly on their modal *frequencies*, should be conducted.
E Process–Product: analyses – solar arrays – frequencies
C Product–Product: solar arrays – frequencies
- Control *algorithms* used to *mitigate* the effects of *solar–array excitations* should be refined.
C Product–Product: algorithm – solar arrays
E Process–Product: mitigate – excitations
- Long *appendages* can deform and cause the *spacecraft* to shiver during eclipse *transitions*.
C Product–Product: appendages – spacecraft
E Product–Process: spacecraft – transitions
- Effective attitude control *algorithms* should be *develop*ed to *analyze* shivering of *spacecraft* during eclipse *transitions*.
C Product–Product: algorithm – spacecraft
E Process–Product: develop, analyze – algorithm
B Process–Process: develop – analyze – transitions

14 Look Beyond Specifications in Qualifying Materials by Similarity

- Substitute *materials* should be *test*ed under conditions that realistically *simulate flight* conditions and give results comparable to those exhibited by the original material.
  E Product–Process: materials – test
  B Process–Process: test – simulate – flight
- The *replace*ment *material* outgassed and delaminated during *firing*. This problem escaped *qualification* since slow heating *rates* (0.1–deg F/sec) used in the lab *test* provided time for the gas to escape. Faster rates would have revealed the issue.
  E Product–Process: material – replace, firing; rate – test
  B Process–Process: replace – qualification – test – firing
- A *supplier* problem prompted the *contractor* to select a *replace*ment *resin* for the *nozzle skirt*.
  A People–People: supplier – contractor
  G People - Process - Product: contractor – replace – resin
  C Product–Product: resin – nozzle skirt
- A rocket *nozzle* failed during *test* firing because a replacement *insulator* delaminated.
  E Product–Process: nozzle – test
  C Product–Product: insulator – nozzle
- The propulsion *valves* in a *rocket* broke down just before *launch* because the oxidizer reacted with a new *cleaning solvent*.
  C Product–Product: valves – rocket
  E Product–Process: solvent – cleaning
  B Process–Process: cleaning – launch
- A *solar array* would not *deploy* in space because *radiation* caused a *rubber spacer* to become sticky.
  C Product–Product: solar array – rubber spacer
  E Product–Process: radiation – deploy

15 Avoid Separable Flared Fittings

- Separable *fittings* in *fluid lines* should be *avoid*ed wherever practical in favor of building permanent connections such as *weld*ed or *braze*d *joints*.
  C Product–Product: fittings – fluid lines – joints
  E Product–Process: fittings – weld, braze
  B Process–Process: avoid – weld, braze
- Where separable connectors must be used, the *fittings* should have machined *sleeves* or redundant sealing *surfaces*. All separable *connectors* should be readily *design*ed accessible at all stages of *building* and at the *launch* site to allow *torque checks* and *repairs*.
  C Product–Product: fittings – sleeves – surfaces – connectors – torque*
  E Product–Process: connectors – design
  B Process–Process: design – building – checks – repairs – launch
- All separable *fittings* should be *torque–check*ed as close to *launch* as possible. If torque checks are not possible within 10 days prior to launch, locking *devices* that do not cause contamination should be used.
  E Product–Process: fittings, torque* – check, launch
  C Product–Product: fittings – devices
  B Process–Process: check – launch
- The flared–fitting *seal* relies on *maintain*ing the required clamping *force* high enough to deform the *flare* into a fit on the *thread*ed elements during *flight*.
  C Product–Product: seal – force* – flare – thread
  E Product–Process: seal – flight
  B Process–Process: maintain – flight

16 Systematically Monitor and Control Contamination

- *Engineers* should *review* the importance of *contamination–control* engineering during every phase of *development* and *hardware design*.
  G People - Process - Product: Engineers – review – contamination*
  B Process–Process: review – control – development
  E Process–Product: design – hardware
  C Product–Product: hardware – contamination*
- Perform contamination *budget analysis*, using *tools* derived from experimental *data*.
  B Process–Process: budget – analysis
  E Process–Product: analysis – data
  C Product–Product: tools – data
- Establish quantitative cleanliness requirements and apply cutting–edge processes to *control* particulate and molecular *contamination*.
  E Process–Product: control – contamination*
- *Contamination* of *radiators* makes *electronics* run hotter.

C Product–Product: contamination* – radiators – electronics

17 Watch Out for the Return of Leonid Micrometeoroid Storms

- *Engineers* and *Satellite Operators* should be *train*ed on the *space environment* situation is vital.
  A People–People: engineers – satellite operators
  G People - Process - Product: Engineers, Satellite operators – train – space environment
- *Satellite operators* should advance *planning* in anticipation of the coming *storms* is essential.
  G People - Process - Product: satellite operators – planning – storms
- *Satellite operators* turning *telescopes* away from incoming *particles*, adjusting *solar panels*, and *orienting* the *satellite* to face the micrometeoroids at an oblique angle minimize *damage* to internal *hardware*.
  G People - Process - Product: satellite operators – damage – hardware
  C Product–Product: telescopes, solar panels, satellite – particles
  B Process–Process: orienting – damage
  E Process–Product: orienting – satellite
- *Satellite operators review procedures* for *rebooting subsystems*.
  G People - Process - Product: satellite operators – review – procedures*
  E Process–Product: rebooting – subsystems
  C Product–Product: procedures* – subsystems
- Making sure experienced *personnel* are *operating* during the *storm*.
  G People - Process - Product: personnel – operating – storm*
- *Satellite operators turn off equipment* that are sensitive to electrostatic discharge (*ESD*), and *avoid* commanding the *satellite* or firing *thrusters* during *storms*.
  G People - Process - Product: satellite operators – turn off – equipment
  B Process–Process: turn off – avoid
  C Product–Product: ESD* – satellite, thrusters
  E Process–Product: avoid – storms*

18 Make Sure Critical Software Performs in its Intended Environment

- *Hardware redundancy* does not necessarily *protect* against *software* faults.
  C Product–Product: hardware – software
  B Process–Process: redundancy – protect
  E Process–Product: redundancy – hardware
- Mission–critical *software* failures should be included in *system reliability* and *fault analysis*.
  C Product–Product: software – system
  E Product–Process: software – reliability, fault analysis
  B Process–Process: reliability – fault analysis
- *Software specifications* should always include specific *operation*al scenarios.
  C Product–Product: software – specification*
  E Product–Process: specification* – operation
- *Software reuse* should be thoroughly analyzed to ensure *suitability* in a new environment, and all associated *documentation*, especially assumptions, should be *reexamined*.
  C Product–Product: software – documentation*
  E Product–Process: software – reuse
  B Process–Process: reuse – suitability, re–examine
- Extensive *testing* should be performed at every level, from *unit* through *system* test, using realistic *operation*al and exception scenarios.
  B Process–Process: testing – operation
  C Product–Product: unit – system
  E Process–Product: testing – unit
- As *software* takes over many *functions* that used to be controlled by *hardware*, *code* sizes increase almost exponentially.
  C Product–Product: software – hardware; software – code
- *Software reliability* thus poses a growing challenge and warrants more *quality assurance* efforts.
  E Product–Process: software – reliability
  B Process–Process: reliability – quality assurance

19 Be Sure that the Architecture Isolates Faults

- Create and use a *verification* matrix for all levels of test *requirements*.
  E Product–Process: requirements – verification
- *Inspect* all test *data* for *trend*s, oddities, and out–of–family values, even when all values are within expectation.
  E Product–Process: data – inspect
  B Process–Process: inspect – trend
- *Evaluate* all indicators for potential impacts, should *trend*s continue. Seek to explain all instances of anomalous *data*.

E Process–Product: evaluate – data
B Process–Process: evaluate – trend

- Incorporate flight *software* into *test* at the earliest opportunity.
  E Product–Process: software – test
- *Avoid* sneak *failure paths* by keeping *circuit design*s straightforward.
  C Product–Product: failure paths – circuit
  E Product–Process: circuit – design
  B Process–Process: avoid – design
- Use isolation *resistors* or downstream *fuses* to prevent a grounded *component* from bringing down the entire *system*.
  C Product–Product: resistors, fuses – component, system

20 Thoroughly Analyze and Test Deployables

- Make sure the *design* can be effectively *test*ed.
  B Process–Process: design – test
- Avoid unconventional *design*s, especially those involving complex *motions*.
  E Process–Product: design – motions*
- Deployable *design* should not be so complex that it cannot be *verified* on the ground.
  B Process–Process: design – verified
- The *deployment* scheme in the *satellite* was too complex to be *test*ed, and The *Aerospace* Corporation had to run an in–depth *analysis* to verify it.
  B Process–Process: analysis – test – deployment
  G People - Process - Product: Aerospace – analysis – satellite
  E Product–Process: satellite – deployment
- Although the *deployment* proved successful in *space*, the *contracto*r learned a lesson and decided to revert to simpler *scheme*s in the future.
  B Process–Process: deployment – scheme
  G People - Process - Product: contractor – scheme – space*

21 Prevent Loss of Lubricating Oil and Grease During Storage and Test

- Use enough *oil* to sustain *storage* and *operation* needs.
  E Product–Process: oil – storage, operation
  B Process–Process: storage – operation
- If porous *hardware* requires *lubrication*, they should be thoroughly *clean*ed, *protect*ed from *moisture,* and *store*d in *oil*.
  C Product–Product: hardware – lubrication* – moisture* – oil
  E Product–Process: hardware – store
  B Process–Process: clean – protect – store
- *Test* high–speed moving *parts* in an inert environment to prevent *oxidation*.
  E Process–Product: test – parts
  C Product–Product: parts – oxidation*
- Perform *materials* compatibility *analysis* to avert *chemical reactions*.
  E Product–Process: materials – analysis
  C Product–Product: materials – reactions*
- Check *NASA* Mechanisms Handbook (NASA/TP–1999–206988) for guidelines on mechanical *assemblies*.
  F People–Product: NASA – assemblies
- The spin *axes* of *gyros* and *wheels* should be oriented during *storage* in such a way as to ensure *oil retention*.
  C Product–Product: axes – gyros – wheels – oil
  E Product–Process: wheels – storage
  B Process–Process: storage – retention
- Minimize *oil* evaporation and migration during *hardware storage*.
  C Product–Product: oil – hardware
  E Product–Process: oil – storage

22 Be Aware of Challenges in Silver/Zinc Battery Manufacturing and Deployment

- *Design, document*ation*, manufacturing, storage,* and *field* application of *batteries* require constant vigilance.
  B Process–Process: design, document, manufacturing, storage, field
  E Product–Process: batteries – design, document, manufacturing, storage, field
- *Materials* must be thoroughly *screened* before being incorporated in *batteries*.
  E Product–Process: materials – screened
  C Product–Product: materials – batteries
- *Batteries* consist of numerous *cells*, each containing a silver *electrode* and a zinc electrode. One of the most common battery problems pertains to the plastic *separators* that wrap around the silver electrodes.
  C Product–Product: batteries – cells – electrode – separators

- Minor *changes* in the constituents of these items have led to incompatibility problems with the *electrolytes*, causing excessive shrinkage or chemical *reactions*.
  E Product–Process: electrolytes – changes
  C Product–Product: electrolytes – reactions*

23 Make Sure Requirements Are Developed Correctly

- Formalize *requirement development* process and capture lessons.
  E Product–Process: requirement* – development
- Provide adequate *design* margins and *operation*al flexibility, such as the ability to use *software patches*.
  E Process–Product: design, operations – software patches
  B Process–Process: design – operations
- Make sure that the *hardware* or *software* which a *contractor* wants to *reuse* from another *program* is indeed applicable and has a satisfactory flight history. Do not be deterred by the excuse that details are not available because the previous program had *proprietary data* or classified—there are always ways to get around that hurdle.
  A People–People: contractor – program
  G People - Process - Product: contractor – reuse – hardware or software
  C Product–Product: hardware – software
  F People–Product: contractor – proprietary data
- Most of the project's *costing* of *performance* is established by front–end *decision*s, but mistakes made there are difficult to catch.
  G People - Process - Product: decision – costing – performance*
- More *resources*, including the most experienced *personnel*, should be made *avail*able to ensure the early *decision*s are made properly.
  A People–People: resources – personnel – decision
  D People–Process: resources, personnel, decision – avail
- *Designers* should thoroughly *review* the history of similar projects. If the *probe designers* had *analyze*d the *requirements* of other deep space projects, both the importance of the *Doppler Shift* and the correct way to perform *end–to–end test* would have become obvious.
  C Product–Product: probe – Doppler Shift*
  G People - Process - Product: Designers – review, analyze – requirements
  E Product–Process: Doppler Shift* – end–to–end test

24 Safeguard Hardware Against Inadvertent Overtesting

- Make sure that test *facilities* are *maintain*ed and *check*ed.
  E Product–Process: facilities* – maintain, check
- Implement over–*test* protection (such as over–temperature tripping *circuits* in *thermal* chambers).
  C Product–Product: thermal* – circuits
  E Product–Process: circuits – test
- Take risks of over–testing during *vibration test*s into account. In particular, large *satellites* should typically be *acoustic*ally tested instead of vibration–tested to prevent damage.
  C Product–Product: vibration*, acoustic* – satellite
  E Process–Product: test – vibration*, acoustic*
- Step up *vibration tests* from one–third to one–half of the full level so that the required *force* can be more accurately computed.
  E Process–Product: test – vibration*, force*
  C Product–Product: vibration* – force*
- Test procedures, set up, and *data* should be thoroughly *check*ed to account for *operator* mistakes and avoid damage.
  G People - Process - Product: operator – check – data
- *Friction* during start–up can greatly exceed that during operation. This problem, known as *stiction*, frequently causes trouble. For example, when a *tape drive* is adjusted, the tape may not move until enough *voltage* to overcome the stiction is applied; but then the force is too large, and the tape suddenly runs wild.
  C Product–Product: friction*, voltage* – tape drive
  E Product–Process: tape drive – stiction

25 Thoroughly Verify All Software Changes

- A small *software* error can have catastrophic *mission* impacts.
  E Product–Process: software – mission
- *Software change* processes require the same degree of rigor as the original *development*. Each change and associated rationale must be individually *approved*.
  G People - Process - Product: approved – change – software
  B Process–Process: change – development
- Retest and regression *test*ing should be formal and thorough. All *logic paths* affected by *change*s must be *verified*, and all results must be checked.

E Process–Product: change – logic path
B Process–Process: change – verified – test

- *Operation*al status, particularly off–nominal *indicators*, must be displayed effectively.
  E Product–Process: indicators* – operation

26 Make Sure Hardware Analyzed Is Hardware Actually Built

- *Designers* should be called back to *inspect* the *products*, to see if there are major differences between *analysis* and *implementation*.
  G People - Process - Product: Designers – inspect – products
  B Process–Process: analysis – implementation
- *Modeling* mistakes are not easily caught. *Analysis* does not negate *testing*.
  B Process–Process: modeling – analysis – testing
- Do not cut corners on *modeling* or *testing*.
  B Process–Process: modeling – testing
- *Programs* should insist that the *analysts document* their methodology and assumptions, and compare them against the actual *hardware* so that errors may be found.
  G People - Process - Product: analysts – document – hardware
  A People–People: programs – analysts
- Do not rely on heritage *design*s until their flight *experiences* are thoroughly understood.
  D People–Process: experiences – design
- *Cells* near the *harness* became hotter and degraded first.
  C Product–Product: cells – harness

27 Control Propellant Balance

- Make sure tank *loads* are *balanced*.
  E Product–Process: loads – balanced
- Use a single *tank*, if feasible, to avoid *propellant migration*.
  C Product–Product: tank – propellant
  E Product–Process: propellant – migration
- Ensure that attitude–control *algorithms* and *mechanisms* can correct dynamic *instability* caused by *propellant* imbalance.
  C Product–Product: instability* – algorithms – mechanisms – propellant
- If possible, *place* a gas pressure *regulator* above the *tanks*, or latching isolation *valves* below each tank, to control *propellant* flow. As *satellites* spin during transfer *maneuvers*, mass imbalances coupled with centrifugal forces can cause tilting. Severe tilt can divert the transfer thrust and prevent *satellites* from reaching their proper *orbit*.
  C Product–Product: regulator, tanks, valves, propellant, satellites, orbit
  E Process–Product: place – regulator
  B Process–Process: place – maneuvers
- Feedback *loops* can be *design*ed to control *gas* pressure or fuel flow between the *tank*s to restore balance. The latter method is more precise.
  E Product–Process: loops – design
  C Product–Product: gas – tanks

28 Graphite/Epoxy Structures Are Easily Damaged by Processing Changes and Handling Mishaps

- Protect graphite/epoxy pressure *vessels* from *handling* damages.
  E Product–Process: vessels – handling
- Insist on *safety* margins and quality *inspections* for composite *structures*.
  E Product–Process: structures – safety, inspections
- Perform extensive *requalification* and acceptance *test*s to guard against subtle processing *changes*.
  B Process–Process: requalification – tests – changes
- In addition to graphite *epoxy*, Kevlar epoxy *structures* are also easily *damage*d.
  E Product–Process: structures – epoxy
  B Process–Process: epoxy – damage
- In both cases, external impact usually leads to *damage* on the inside and can be difficult to *detect*.
  B Process–Process: damage – detect

29 Validate Changes in Command Script Configuration

- Treat command–Procedure *changes* with the same rigor as flight–critical *software*. This includes formal *configuration management*, peer *review* with knowledgeable technical *personnel* and full command *verification* with an up–to–date *simulator*.
  C Product–Product: software – simulator
  G People – Process - Product: personnel – changes, CM, review, verification – software
- Ensure *change* implementation timelines are consistent with *staff* workloads.

D People–Process: staff – change

- Display *spacecraft health* and *safety* information clearly.
  E Product–Process: spacecraft – health, safety
- Follow *validate*d *operations procedures*, including *review* of all pertinent *data*.
  C Product–Product: data – procedures*
  E Process–Product: validate, review, operations – data, procedures*
  B Process–Process: validate – review – operations

30 Maximize On–board Reprogrammability to Enable Fault Recovery

- *Design* into the *satellite* the flexibility to *handle* unforeseen emergencies, and provide emergency reset capability for major *components*.
  E Process–Product: design – satellite
  B Process–Process: design – handle
  C Product–Product: satellite – components
- *Add* emergency *protection* of a *satellite battery* system, such as low–battery–voltage cutout of nonessential *loads*.
  E Process–Product: add – protection*
  C Product–Product: satellite – battery – loads
- The fuel *tank* had to be warmed up before *pipes* and *thrusters* were, lest overpressure burst the lines. *Software changes* allowed the *battery* to discharge current like a *thermistor* and turn on selective *heaters* whenever power became available. Because the flight *computer* was off during *battery* charging, the software patch had to be reloaded each time. After fine–tuning, *controllers* managed to thaw the *tanks* with 48 *heaters*, using a peak power of over 500 watts!
  C Product–Product: tank – pipes – thrusters; software – battery; thermistor – heaters; computer – battery; controllers – tanks – heaters
  E Process–Product: change – software

31 Oxidation Can Cause Erratic Open Circuits In Solid State Devices

- Protect sensitive *metal* layers from oxidation (caused by over–etching, for example) during *semiconductor fabrication*.
  E Product–Process: metal – fabrication
  C Product–Product: metal – semiconductor
- Use current–voltage *profiles* as a *diagnostic* tool—nonlinear high resistance usually indicates oxidation.
  B Process–Process: diagnostic – profiles
- An applied *voltage* can sometimes heal the chips temporarily by pushing the oxide *layer* aside.
  C Product–Product: voltage – layer
- Incomplete coverage of the *gold* via by the trace exposed the *titanium* layer to inadvertent oxidation.
  C Product–Product: gold – titanium

32 One Operation, One Verification

- Implement a discrete *verification* step for each critical *task*.
  B Process–Process: verification – task
- *Avoid* multiple *tasks* within a *procedure*.
  B Process–Process: avoid – task
  E Process–Product: task – procedure*
- Ensure a *fail–safe* process by applying *software* technology, self–checking indicators, or positive feedback mechanisms to complex *operations* vulnerable to *human* errors.
  E Process–Product: fail–safe – software
  G People – Process - Product: human – operations – software
- *Document* each near miss and correct its *root cause*.
  B Process–Process: document – root cause
- The precision *regulator* in a *booster engine* control system used a *stem screw* to modulate *gas inlet*. A set screw forced a nylon *plug* against the stem screw *threads* and prevented the stem from rotating. The regulator was reworked to *repair* leakage during build. The *rework instruction* did not explicitly require set screw *re–torquing* and *verification*. The loose set screw caused the stem screw to unseat. The *launch* failed.
  C Product–Product: regulator – booster engine; screw – gas inlet; plug – threads
  E Product–Process: regulator – repair; instruction – rework
  B Process–Process: repair – verification – launch

33 Check Satellite–Launcher Compatibility As Early As Possible

- Ensure interface problems between the *satellite* and *launcher*, such as dynamic *instability*, are analyzed early on in the *design* process.
  C Product–Product: satellite – launcher
  E Product–Process: instability* – design
- Solid upper *stages*, which are used in a *mission*, are more prone to *instability*.

E Process–Product: stages – instability*
B Process–Process: stages – mission

- The *satellite contractor* did not recognize this risk in part because the *launch vehicle contractor* failed to formally *communicate* this *requirement*.
  A People–People: satellite contractor – launch vehicle contractor
  G People - Process - Product: contractors – communicate – requirement
  C Product–Product: satellite – launch vehicle
- The *design changes* kept the *instability* in check during *flight*, and the *satellite* reached the correct orbit.
  E Process–Product: design – instability*
  C Product–Product: instability* – satellite
  B Process–Process: design – changes – flight

34 Safeguard Hardware Against Inadvertent Overtesting (II)

- Implement over–*test protection*.
  E Process–Product: test – protection*
- *Correct* the *root cause* of operational *mistakes*.
  D People–Process: mistakes – root cause
  B Process–Process: correct – root cause
- Incorporate visual guides or *overlays* as part of process *control* procedures.
  E Product–Process: overlays – control

35 Implement Independent Fault Protection

- Apply independent fault *protection* for critical *software* functions.
  C Product–Product: protection* – software
- Implement exception *handling* to *protect* the flight *processor* from aborts due to *data* handling errors.
  C Product–Product: protect* – processor – data
  E Process–Product: handling – data
- Do not cut corners in *testing* critical flight *software*.
  E Process–Product: testing – software
- Over 65,000 lines of flight *code* (only 20% inherited) were developed in 17 *increments* within one year, leaving little time for thorough *testing*.
  E Product–Process: code – increments
  B Process–Process: increments – testing

36 Implement Independent Fault Protection (II)

- Create extensive, realistic nominal and anomalous operational *scenarios* for *testing* at every level, from *unit* through *system* test.
  E Product–Process: scenarios* – testing
  C Product–Product: unit – system
- Implement robust *simulators*, including *hardware*–in–the–loop, for *testing* critical flight *software* functions.
  C Product–Product: hardware – software
  E Process–Product: testing – software
  B Process–Process: simulators – testing
- Apply independent fault *protection*, such as *hardware* watchdogs, to *mitigate* risk in realtime *systems*, where errors can be so deeply buried as to be practically *undetect*able.
  C Product–Product: protection* – hardware – system
  B Process–Process: mitigate – undetect
  E Process–Product: mitigate – system
- The processor feeds a series of programmed *pulses* into the hardware *timer*, which will reset itself and await the next input. If the expected heartbeat does not arrive, the watchdog knows that the *processor* has probably crashed and intervenes (such as by initiating a fault protection *routine*).
  C Product–Product: pulses – timer; processor – routine

37 Aim for Realistic Schedules in Development Projects

- Provide a detailed interface *specification* as early in the *system lifecycle* as possible.
  E Product–Process: specification – lifecycle
  C Product–Product: specification – system
- *Program Office* fosters a cooperative working arrangement among *system contractors* and proactively *maintains* realistic *power, weight,* and *volume* reserves.
  A People–People: program office – contractors
  G People – Process - Product: contractors – maintain – power*, weight*, volume*
  C Product–Product: power*, weight*, volume* – system

- Create engineering *model*s so that problems can be *discover*ed early.
  B Process–Process: models – discover
- Slim *margins,* unproven *technology,* tight *schedules,* and fixed *cost* conspired to incrementally push the *delivery* date.
  E Product–Process: margins, technology, schedule, cost – delivery

38 Do Not Ignore Unexplained Test Anomalies

- *Test* under all operating conditions—not only sunlight and eclipse *operation*, but *transitions,* safe–hold mode, load–shed mode, and recovery *mode*.
  B Process–Process: test – operations, transitions, mode
- Strive to *understand* implications of *test* anomalies.
  D People–Process: understand – test
- Ensure perceptive instrumentation, lest *test*–set glitches cast *doubt* on results.
  D People–Process: doubt – test
- Minor *design* changes in *power supplies* can result in disastrous consequences. Double–*check* design *changes*, and perform independent *analysis* where practical.
  E Process–Product: design – power supplies
  B Process–Process: check – changes – analysis
- The line *filter* and feed–through *capacitor combine*d to resonate at a crossover *frequency*. The *array* would suffer *sustain*ed oscillation and fail.
  C Product–Product: filter – capacitor – array
  E Process–Product: combine – frequency*
  B Process–Process: combine – sustain

39 Thoroughly Review Test Data for Early Indicators of Anomalies

- Carefully inspect all *test* and operational *data* for *trend*s, oddities, and out–of–family values, even when all values are within preset limits. *Evaluate* all indicators for potential impacts, should trends continue. Seek to *explain* all instances of anomalous data.
  B Process–Process: test – trends, evaluate, explain
  E Process–Product: test – data
- Make sure that experienced *operators* closely monitor the *satellite*'s *health* during early operations.
  F People–Product: operators – satellite
  C Product–Product: satellite – health
- Provide *ground*–commandable *back–up heaters*.
  C Product–Product: ground – heaters
  E Process–Product: back–up – heaters
- Install *heaters* to fill/drain *lines*, and provide temperature *monitors* for all propellant lines and *valves*.
  C Product–Product: heaters, monitors – lines, valves
- Damage of the *wiring* at the heater *lead* probably caused the failure. A more robust *configuration* was used in all subsequent *flights*.
  C Product–Product: wiring – lead
  B Process–Process: configuration – flights
  E Process–Product: configuration – wiring
- Although the *heater* failed during early ground *tests*, the problem was not recognized because temperature limit *checks* were set to accommodate test environment *changes*, not to verify heater performance. Later *tests* and *operations* used *computer*–controlled stepwise limit checks to highlight anomalous *behaviors* early.
  E Product–Process: heater – test
  C Product–Product: computer – behavior
  B Process–Process: checks – changes; test – operations

40 Avoid Radio Frequency Interference

- *Understand* why *requirements* exist in legacy *design*s before discarding them.
  G People – Process - Product: understand – design – requirements
- Coordinate *spectrum planning* with *authorities* (for example, Manager of Spectrum Allocation at the Space Command), because not all frequency usages are *public* information.
  G People – Process - Product: authorities – planning – spectrum
  A People–People: authorities – public
- *Emission* from *crosslinks* can reach Earth and interfere with other *users*.
  G People – Process - Product: users – emission – crosslinks
- The *emission* problem can be cured by phasing the *signals* in the *array* to place a *null* toward Earth.
  C Product–Product: signals – array
  E Product–Process: null – emission

41 Carefully Consider the Implication of Test Failures Beyond the Narrow Issues at Hand

- Thoroughly *evaluate* the heritage and applicability of using existing or flight–Proven *equipment*, especially if *modifications* have been made.
  E Process–Product: evaluate – equipment
  B Process–Process: evaluate – modifications
- Include shorting in analyzing potential failure *modes* of power *systems*.
  E Process–Product: analyzing – modes
  C Product–Product: modes – systems
- Apply *manufacturing* and *handling* practices that minimize slip *ring damage*.
  B Process–Process: manufacturing – handling – damage
  E Process–Product: damage – ring
- Shorting of slip *rings* is fairly common— improperly lubricated *brushes* can easily abrade conductive slivers out of the rings.
  C Product–Product: rings – brushes
- The voltage gap across adjacent *brushes* exacerbated shorting by triggering an arc, which wrecked every *anode* in its path.
  C Product–Product: brushes – anode

42 Account for Electrostatic Interaction in Structural Analysis

- Be aware of the propensity of *dielectrics* to pick up an *electrostatic* charge in *space*.
  C Product–Product: dielectrics – electrostatic*
  E Product–Process: dielectrics – space
- Thoroughly *review* the potential *impacts* of the *space environment* on flight *hardware*.
  B Process–Process: review – impacts
  E Product–Process: hardware – space environment
- Whenever possible, a *design*'s operation in space (0 G) should be designed to be *verifiable* under 1 G test *conditions*.
  B Process–Process: design – verifiable
  E Process–Product: design – conditions*
- *Test* the entire *system* in the final flight *configuration*.
  E Process–Product: test – system
  B Process–Process: test – configuration
- The *sunshield* curled toward the *antenna* due to charges that accumulated in the *insulators*. Notice that *electrostatic* attraction can take place even though one surface (the *sunshield* in this case) is grounded.
  C Product–Product: sunshield – antenna – insulators – electrostatic*

43 Do Not Circumvent Processes Designed to Catch Human Errors

- Ascertain software *databases* as thoroughly as the *source codes*.
  C Product–Product: databases – source code
- *Verify software algorithm* and *database* on a *simulator* whenever possible.
  C Product–Product: algorithm – database
  E Product–Process: software – simulator
  B Process–Process: verify – simulator
- Double–*check* manually entered *data* against original *sources*.
  E Process–Product: check – data
  C Product–Product: data – sources
- *Automate data* transfer and checking whenever possible to minimize *human* error.
  G People – Process - Product: human – automate – data
- *Programmer applied* an incorrect *formula* in the ground *software* led to the *failure* of *Mariner I* in 1962.
  C Product–Product: formula – software; software – Mariner I
  G People – Process - Product: programmer – applied – software
  E Product–Process: software – failure

44 Beware of Sneak Paths Through Test Equipment

- *Determine* and *correct* the root cause of all failures.
  B Process–Process: determine – correct
- *Trace* the flow of *power* and *signals* from source to load during *troubleshooting*.
  E Process–Product: trace – power, signals
  C Product–Product: power – signals
  B Process–Process: trace – troubleshooting
- Provide a *mechanism* to independently *validate* the status of critical *components*.
  B Process–Process: mechanism – validate
  E Process–Product: validate – components
- Inject unexpected *conditions* (such as a closed relay, current surge, and sluggish separation wire breakage) during reliability *analysis* to discover lurking failure paths.

E Product–Process: conditions* – analysis

- A *latch* in the separation *sensor* (powered via *relay*) opens after the *satellite* breaks away from the *launcher*, deploying the *solar array* via relay.

  C Product–Product: latch – sensor – relay – satellite – launcher – solar array
- Failure of *relay*, due to the addition of a *filter*, formed a sneak path (dashed line) via the simulator *port*, triggering the *prelaunch* anomaly. Premature separation in fact could not occur in *flight* because the port is not used.

  C Product–Product: relay – filter – port

  E Product–Process: port – prelaunch

  B Process–Process: prelaunch – flight

45 Guard Against Chloride Contamination Due to Manufacturing Process Changes

- Heat *pipes* are highly sensitive to minor *materials* and process *changes*.

  C Product–Product: pipes – materials

  E Product–Process: pipes – changes
- Seemingly minor process *alterations* can have catastrophic side *effects*.

  B Process–Process: alterations – effects
- Allow sufficient *time* before conducting *tests* of *chemical degradation.*

  B Process–Process: time – test
- An *engine* suffered severe leak during recent ignition *testing* because the *chamber* was *cleaned* with over–the–counter detergent.

  E Product–Process: engine – testing

  C Product–Product: engine – chamber

  B Process–Process: testing – cleaned
- *Chloride* in the *cleaner* induced stress *corrosion*, cracking the *tubes*.

  C Product–Product: chloride – tubes

  B Process–Process: cleaner – corrosion

  E Process–Product: cleaner – tubes

46 Make Sure Test Equipment Is Sufficiently Capable

- *Budget* for high fidelity, reproducible, functional *tests* to facilitate troubleshooting.

  B Process–Process: budget – test
- *Troubleshooting* was hampered because the test set could not monitor all *channels*.

  E Process–Product: troubleshooting – channels
- The *reliance* on *oscilloscopes* made *data collection* inefficient.

  B Process–Process: reliance – collection

  E Process–Product: reliance – oscilloscopes, data

  C Product–Product: oscilloscopes – data
- Digital *data collection* from all *ports solve*d the problem in a few days.

  E Process–Product: collection – data

  B Process–Process: collection – solve

  C Product–Product: data – ports
- *Housekeeping* (as opposed to hardware–related) glitches in *facility*, *software*, *equipment*, or *connectors* routinely account for the majority of discrepancy reports, unnecessarily impacting program scheduling.

  B Process–Process: housekeeping – scheduling

  C Product–Product: facility – software – equipment – connectors

  E Process–Product: housekeeping – software

47 Review Hardware Reusability When Configuration Changes Affect Margins

- Recognize that workmanship plays a large role in the space *hardware*, and *reliability* may be compromised when undertrained *personnel assemble* heritage equipment.

  G People – Process - Product: personnel – assemble – hardware

  B Process–Process: assemble – reliability
- Computerize manufacturability *analysis*, including interface tolerance buildup, dynamic interference, and ease of *inspection* on all *packaging designs*.

  B Process–Process: design – analysis – inspection – packaging
- Provide *automatic* fault *management* mechanisms so that a single defect will not bring down the entire system.

  B Process–Process: automatic – management
- An *inspection* of the *hardware* destined for the next flight revealed that many *screws* were too long to fit into the space between the relay *mount* and the radiator *plate*, making a short virtually inevitable.

  E Process–Product: inspection – hardware

  C Product–Product: screws – mount – plate
- Moreover, the *heatsink* barely cleared the unit *walls*. Because the heatsink was not conformably *coated*, debris such as a loose

solder ball could also have caused a short.
C Product–Product: heatsink – walls
E Process–Product: coated – heatsink

48 Thoroughly Reverify Software When Requirements Change

- *Re–verify software* performance when it's intended environment *changes*.
  E Process–Product: re–verify – software
  B Process–Process: re–verify – changes
- Thoroughly *analyze* the impact of loss of *precision*.
  E Process–Product: analyze – precision*
- Ensure change *analysis* is complete and *changes* are comprehensively *verified*.
  B Process–Process: analysis – changes – verified
- Cumulative *precision* loss let the *radar* look in the *wrong place* (*range gate*) for the *Scud*.
  C Product–Product: precision* – radar; radar – range gate – Scud

49 Equipment Intended for Use in Simulated Space Environments Should Be Space–Rated

- Perform formal design *review*s on ground–test *equipment* intended for use in space–like environments.
  E Process–Product: review – equipment
- *Test* radio frequency *equipment* in *vacuum* to 6 decibels over the expected input level (to account for unfavorable signal return) to ensure operational *safety*.
  C Product–Product: equipment – vacuum
  B Process–Process: test – safety
  E Process–Product: test – equipment
- Monitor flight *hardware* during *test* lest overstressing cause *damage*.
  E Product–Process: hardware – test
  B Process–Process: test – damage
- Improve interfaces between *payload engineers* and *bus engineers*, particularly during *system* level *tests*.
  A People–People: payload engineers – bus engineers
  G People – Process - Product: engineers – test – system
  C Product–Product: payload – bus – system
- A test set scheduled for use in the thermal vacuum *chamber* contained cadmium–Plated parts. *Cadmium*, commonly used to plate military *components*, sublimes in vacuum and is not allowed in *space*. If the test had gone ahead, the cadmium could have contaminated not only the *spacecraft* being *test*ed, but also the chamber and future satellites!
  C Product–Product: chamber – cadmium; chamber – spacecraft; component – spacecraft
  E Product–Process: cadmium – test
  B Process–Process: test – space

50 Virtual Cross–strapping Extends Satellite Life

- On–board *reprogrammability* provides enormous *flexibility*.
  B Process–Process: reprogrammability – flexibility
- In a tight spot, seek *cross–Program* wisdom from diverse *organizations*.
  A People–People: cross–Program – organizations
- Capture *knowledge* of heritage designs and look for novel ways to take advantage of *design* features.
  D People–Process: knowledge – design
- The *gimbal* controller design included a path to forward–control nonlinear *motor* driver behavior.
  C Product–Product: gimbal – motor
- The rescue *scheme* fed commands, derived from *sensor* A data and calculated by the *processor* using new control laws, into the *motor* controller B via this route, bypassing the processor B.
  E Process–Product: scheme – sensor – processor – motor

51 Review Troubleshooting Process When Encountering Surprising Test Results

- Consider using *bar coding* in *production* control.
  B Process–Process: bar coding – production
- Incorporate *design* features, such as colored *cables*, to preclude *human* errors.
  G People –Process - Product: human – design – cables
- Don't *overlook* simple *human* errors when confronting unexplained *problems*.
  G People – Process - Product: human – overlook – problems
- A thermal vacuum *test* was delayed because two rolls of Kapton *tapes* were *mixed up*.
  B Process–Process: test – mixed–up
  E Product–Process: tapes – mixed–up
- Both rolls of *tape came from* the same *supplier* and *looked* exactly the *same*.
  G People – Process - Product: supplier – came from – tape

B Process–Process: came from – looked same

- However, the roll of *tape* inadvertently used to attach insulation *blankets* contained an *adhesive* that was based on *silicone* instead of on low–outgassing *acrylics*.
  C Product–Product: tape – blankets – adhesive – silicone – acrylics
- The *satellite* had to be *baked* and *pumped* for a long time before *silicone* outgassing subsided.
  E Product–Process: satellite – baked, pumped
  B Process–Process: baked – pumped
  C Product–Product: satellite – silicone

52 Protect Cryogenic Systems Against Thermal Expansion Mismatch

- Perform in–depth *modeling* and thermal cycling *tests* on cryogenic *systems*, which are delicate *equipment* involving complex physics and material *behavior*.
  B Process–Process: modeling – test
  E Process–Product: modeling, test – systems
  C Product–Product: systems – equipment – behavior*
- Provide adequate *tolerances* for thermal expansion *mismatch* (using flexible links, for example).
  C Product–Product: tolerances* – mismatch*
- Be extra *vigilant* when *stretching* the *state–of–the–art*.
  G People - Process - Product: vigilant – stretching – state–of–the–art
- Because the *aft part*, though which supercold *helium* was pumped, was colder than the *forward part*, forward *nitrogen* could sublime and refreeze aft, eliminating ullage space. After helium flow stopped, the tank warmed up. The large CTE differential (700 ppm/°K for solid nitrogen, 17 ppm/°K for aluminum) probably forced the *dewar* to yield. Progressive deformation gradually closed the gaps between the *baffles*.
  C Product–Product: aft part – helium – forward part – nitrogen – dewar – baffles

53 Test Hardware and Software Together

- Rigorously *control configuration*, especially at *hardware/software* interface.
  E Process–Product: configuration control – hardware, software
  C Product–Product: hardware – software
- Always ascertain torquer polarity.
  E Process–Product: ascertain – polarity*
  C Product–Product: torque – polarity*
- Provide sufficient ground *station coverage* in early *operation*.
  C Product–Product: station – coverage*
  E Product–Process: coverage* – operation
- *Design battery* protection to keep the *satellite* alive long enough for *troubleshooting* by implementing automatic load shedding and by configuring *solar panels* so that even a partially *deployed array* could keep battery charged.
  E Process–Product: design – battery
  C Product–Product: battery – satellite – solar panels – array
  B Process–Process: design – troubleshooting – deployed
- Magnetic *torquers* are *coils* wound around an iron *core*. Passing a *current* through the coils creates a magnetic *dipole* which interacts with the Earth's magnetic field and generates a feeble *torque*. Reversing the current flow (phase) produces the opposite effect. Torquer *polarity* mistakes occur often.
  C Product–Product: torquers – coils – core – current – dipole – torque – polarity
- The orientations of large *coils* are easily *verified* with a magnetometer (essentially a compass). Background noise can make checking small *torquers* difficult.
  C Product–Product: coils – torquers
  E Process–Product: verified – coils, torquers

54 Design and Handle Cryogenic Equipment with Great Care

- *Review* and *follow* operating and transportation *procedures* associated with cryogenic *equipment* to ensure *safety* to *personnel*, flight *hardware*, or *facilities*.
  B Process–Process: review – follow – safety – facilities
  G People – Process - Product: personnel – safety – procedures*
  C Product–Product: procedures* – equipment – hardware
  E Process–Product: review, follow – procedures*
- Provide a graceful failure *mechanism*, if possible, to *prevent* catastrophic failure.
  B Process–Process: mechanism – prevent
- *Design* for containment making sure the *cryogens* that unexpectedly boil off can be constrained within the *vessel*.
  E Process–Product: design – cryogens
  C Product–Product: cryogens – vessel
- Provide *redundant vent* paths.

E Process–Product: redundant – vent

- *Design* for convenient *disassembly* to aid *inspection* and *maintenance*.
  B Process–Process: design – disassembly – inspection – maintenance
- *Service* absolute pressure *valves* often, but never exceed *vendor specifications*. *Test* valves before every field *operation*.
  E Process–Product: service, test – valves
  F People–Product: vendor – specifications
  B Process–Process: service – test – operation

55 Do Not Dismiss Test Anomalies as Random Events–Find Out Why (I)

- Exhaustively *search* for the *root cause* of failures.
  B Process–Process: search – root cause
- Conduct fully *instrument*ed *tests*.
  E Product–Process: instrument – tests
- Provide sufficient thermal and structural *margins* to allow for material, *manufacturing*, and processing fluctuations.
  E Product–Process: margins* – manufacturing
- The independent *investigation* prompted *NASA* to conduct its own instrumented firing, which proved the buckling *scenario*.
  G People – Process - Product: NASA – investigation – scenario*

56 Do Not Dismiss Test Anomalies as Random Events–Find Out Why (II)

- Define and implement a *verification plan*.
  B Process–Process: verification – plan
- Perform a worst–case circuit *analysis* to meet defined interface *requirements*.
  E Process–Product: analysis – requirements
- Always *ascertain* the root causes of ground *test* anomalies.
  B Process–Process: ascertain – test
- Signals from the Sun *Sensor* passed through the EMI *filter*, the slip *rings*, and the *amplifier* to the *controller*. The controller oriented the *boom* by alternating the *motor* between two states (A, A' transistors on the H–bridge open, B, B' closed; and B, B' open, A, A' closed).
  C Product–Product: sensor – filter – rings – amplifier – controller – boom – motor
- The grounded EMI *filter*, coupled with a *circuit* not *design*ed for fast switching, allowed transient noises from the *chassis* to momentarily turn all *transistors* on, blowing the *fuse*. *Installation* of a *resistor* eliminates the noise problem.
  C Product–Product: filter – circuit – chassis – transistors – fuse – resistor
  E Process–Product: installation – resistor

57 Protect Propulsion System from Contamination

- Consider retrofitting legacy *hardware* with proven *design upgrade*s. Anticipate out–of–sequence *operations*, such as rework, during hardware design.
  B Process–Process: design – upgrade – operations
  E Process–Product: upgrade, design – hardware
- *Design* propulsion *systems* to accommodate ground *handling* by including features such as low point *drains* to facilitate *fuel removal*.
  E Process–Product: design – systems
  B Process–Process: design – handling – removal
  C Product–Product: systems – drains – fuel
- *Archive manufacturing* documents.
  B Process–Process: archive – manufacturing
- The higher location of the fill/drain *port* in the legacy propulsion *system* prevents gravity draining, and the single seat *valve* is prone to leak. Dual seat valves, typically used in new *design*s, would have prevented air ingression unless both valves leaked.
  C Product–Product: port – system – valve
  E Product–Process: valve – design
- An *ICBM*, refurbished to *launch satellites*, suffered performance degradation recently after its turbine *seal* leaked, allowing ammonia in the exhaust gas to react with the *lubricant*, plugging the *filter* and blocking lubricant circulation. The problem, chemically alike the *thruster* contamination, was addressed in the follow–on generation of the *rockets*, but the original *units* were not *retrofit*ted.
  C Product–Product: ICBM – satellites – seal – lubricant – filter – thruster – units
  E Process–Product: retrofit – ICBM
  B Process–Process: retrofit – launch

58 Guard Against Sneak Paths Through Ground Test Equipment

- Independently confirm *hardware* performance for functions temporarily provided by *test* equipment.
  E Process–Product: test – hardware
- Use a breakout *box* to *check harness* connector paths, and directions and magnitudes of currents flows.

E Process–Product: test – harness
C Product–Product: hardware – harness

- A flight *box* was not grounded by mistake. The problem was missed because the *test* equipment supplied *grounding*.
  C Product–Product: box – grounding*
  E Process–Product: test – grounding*

59 Lesson from Challenger: Understand Your Data!

- *Consider* all relevant *information*.
  E Process–Product: consider – information
- *Develop* a coherent explanation of engineering *data* to help audience *analyze* risks.
  B Process–Process: develop – analyze
  E Process–Product: develop, analyze – data
- *Display data* cogently.
  E Process–Product: display – data
- A table of temperature *data* presented during *pre–launch teleconference* included irrelevant information but only selective *flight*. The *audience* was misled.
  B Process–Process: pre–launch – flight
  F People–Product: audience – data
  A People–People: teleconference – audience
- Anomalies rarely occurred in *warm days*, but routinely took place during *launch*es below 65°F.
  E Process–Product: launch – warm days*

60 Tests Are for Verification, Not for Discovery

- Expected *test* results should be established in advance of the test. *Deviation* from expected results should raise a flag, and be thoroughly investigated before making any changes.
  B Process–Process: test – deviation
- Rigorously *manage software development*, especially on *requirements*, interfaces, and *configuration control*.
  B Process–Process: manage – development – configuration control
  E Process–Product: development – software, requirements
  C Product–Product: software – requirements
- *Plan* for *contingencies*, using a top–down fault tree (ask "what happens if the satellite failed to de–spin?" for example).
  E Process–Product: plan – contingencies*
- Double–*check torquer* signs.
  E Process–Product: check – torque
- Opposite magnetic *poles* attract. The north pole of *magnet needles* points to the *Earth*'s magnetic. South Pole also called the geomagnetic North Pole!
  C Product–Product: poles – magnet – needles – Earth

61 Do Not Assume a Situation Is Acceptable Simply Because Nothing Is Said About It in Documents

- Double–*check designs* against possible mis–*installation*.
  B Process–Process: check – design – installation
- Make sure field–*assemble*d *hardware* can be *inspect*ed.
  B Process–Process: assemble – inspect
  E Process–Product: assemble – hardware
- An on–board video *camera* captured the inter–stage hang–up, enabling the investigation *team* to create a dynamic *model* and to replicate the problem on a mockup.
  G People – Process - Product: team – model – camera

62 Test as You Fly

- *Analyze* prior incidents of *equipment* malfunction.
  E Process–Product: analyze – equipment
- *Review* all aspects of *battery* application. Do not regard batteries as simple *Plug–and–Play* items.
  E Process–Product: review – battery, Plug–and–Play*
  C Product–Product: battery – Plug–and–Play*
- Dry silver/zinc *batteries* are activated by adding *electrolytes* in a vacuum environment. Once filled, internal reactions can lead to frothing and spattering. *Launch* depressurization and continuous discharging heat up the *cells*, causing more spills.
  C Product–Product: batteries – electrolytes – cells
  E Process–Product: launch – batteries
- Serious mishaps had occurred, even on the ground. Several years ago, a *launch* delay caused a *battery* to exceed its wet life. Days later, it caught fire. Apparently, drops of escaped *electrolyte* made their way along the power *wires* via *capillary* action, shorting a *connector*.
  E Process–Product: launch – battery

C Product–Product: battery – electrolyte – wires – capillary – connector

63 Verify Field Installations of All Single–Point–Failure Items

- *Simplify* interfaces, commands, and *procedures* in *prelaunch operations* lest the hectic pace cause errors.
  E Process–Product: simplify – procedures*
  B Process–Process: simplify – prelaunch – operations
- *Verify* final *assembly operations*, particularly on single–Point–failure risks. Pay particular attention to possible *connector* mis–mating.
  B Process–Process: verify – assembly – operations
  E Process–Product: verify – connector
- Do not *allow* primary and redundant *sides* of critical *circuits* to join in a single–Point failure *area*.
  E Process–Product: allow – sides
  C Product–Product: sides – circuit – area

64 Review Out–Of–Flow Processes to Ensure No Steps Are Bypassed

- Make sure *corrections* in engineering *drawings* or work *instructions* are back annotated in all applicable drawings and shop orders (including subsequent *builds* and *units* that have been distributed).
  E Process–Product: corrections – drawings, instructions
  B Process–Process: corrections – builds
  C Product–Product: drawings, instructions – units
- Conduct final *walkthroughs* in the presence of the most experienced *personnel*.
  D People–Process: personnel – walkthroughs
- Keep good *records* of all "non–flight" *installations*.
  B Process–Process: records – installations
- A *satellite* used active *louvers* to control the *base–Plate* temperature of an *instrument*.
  C Product–Product: satellite – louvers – base–Plate – instrument
- The *system*, including the *louvers*, underwent thermal vacuum *test*ing, after which the louvers were removed. They were temporarily *reinstall*ed, without being connected, for fit check.
  E Process–Product: test – system
  C Product–Product: system – louvers
  B Process–Process: test – reinstall
- The *louvers* were left in place, without anyone realizing that the *connector* remained unattached. Pre–shipment *checks* did not verify the mate status because the connector was not *accessible*.
  C Product–Product: louvers – connector – accessible*
  E Process–Product: check – connector
- Running too hot in *space*, the *instrument* suffered significant degradation.
  C Product–Product: instrument – space*

65 Perform Thorough Post–Flight Analysis

- *Track* down the root causes of *anomalies* and *consider implications* beyond the narrow issues at hand.
  E Process–Product: track – anomalies; consider – implications*
  B Process–Process: track – consider
- Unexpected *hardware* behavior implies a failure to *understand* the application. *Safety* cannot be inferred just because the *mission* succeeded since the problem may be much more severe next time.
  G People – Process - Product: understand – mission – safety*
- Misleading instructions on *drawings* led *assemblers* to *wrap* thermal *tapes* too close to a separation *connector*. The *stage* jammed, stranding the *satellite*.
  G People – Process - Product: assemblers – wrap – tape
  C Product–Product: drawings – tape – connector – satellite
  B Process–Process: wrap – stage
  E Process–Product: wrap – tapes
- Eleven previous *flights* were subsequently *review*ed; all showed the same hang–up.
  E Process–Product: review – flights
- Seven, in fact, were saved only because the floating *connectors* were jolted apart when they hit the allowable *stops*. The *mission* right before the failure had the narrowest escape.
  E Process–Product: stops – connectors
  B Process–Process: stops – mission
- The warning *signs* were not *pursue*d.
  E Process–Product: pursue – signs

66 Thoroughly Analyze All Environmental Load Paths and Develop a Detailed System Dynamic Model

- Provide extra *margins* to accommodate excessive *launch* shocks that occasionally occur, especially with new launch *vehicles*.

E Process–Product: launch – margins*
C Product–Product: vehicle – margins*

- Independently *review* dynamic loads *analysis* prior to *test*.
  B Process–Process: review – analysis – test
- Adequately instrument the *unit*, *subsystem*, and *vehicle* during environment *tests*.
  C Product–Product: unit – subsystem – vehicle
  E Product–Process: vehicle – tests
- *Check* all *data* and inspect critical *parts* for damage after *tests*.
  E Process–Product: check – data; inspect – parts
  B Process–Process: check – inspect – tests

67 Provide Design Flexibility to Enable Emergency Recovery

- Provide as much *telemetry* as possible on *launch vehicles*, especially on separation events. Without knowing how the *satellite* malfunctioned, *controllers* would likely have given up before the downlink was *receive*d!
  E Process–Product: launch – vehicle, satellite
  G People - Process - Product: controllers – receive – telemetry
  C Product–Product: vehicle – satellite – telemetry
- Accurate *attitude knowledge*, especially during orbit night when most of the *observations* were made, posed the next challenge. The *satellite* no longer rotated as a rigid body; even the spin *axis* orientation was uncertain.
  G People - Process - Product: knowledge – observations – attitude*
  C Product–Product: satellite – axis
- The *program* created a non–linear rigid body *model*. Using Sun sensor and horizon crossing indicator *data* as input, an algorithm incorporating Kalman *filters* calculated the *satellite attitude* to 0.25º accuracy, even during most of the orbit nights when direct *sensor* readings were unavailable.
  G People - Process - Product: program – model – attitude*
  C Product–Product: data – filters – satellite – sensor
- Most *mission requirements* were met.
  E Process–Product: mission – requirements*

68 Insist On End–to–End Ownership to Verify Interfaces

- *Develop* end–to–end *diagrams* for *electrical* and *mechanical* interfaces, including *software* driven interfaces.
  E Process–Product: develop – diagrams*
  C Product–Product: electrical – mechanical – software
- Clearly *label* each *connector* to avoid mis–mating.
  E Process–Product: label – connector

69 Protect Solid Rocket Grain Structure from Destabilizing Gas Flow

- *Conduct* adequate sub–scale *testing*.
  B Process–Process: conduct – testing
- *Study* post–*test* and post–*flight* anomaly *reports* from similar programs.
  B Process–Process: study – test – flight
  E Process–Product: study – reports
- The original *design* constricted flow at the segment *joint*. The *grain* cracked, further raising *chamber* pressure.
  E Process–Product: design – joint
  C Product–Product: grain – chamber
- *Chamfering* of the forward *grain* face eliminated the chokepoint.
  E Process–Product: chamfering – grain

70 Late Modifications Require Careful Revalidation

- Perform thorough *analysis* and *testing* of late *hardware changes*. Pay particular attention to *system*–level impacts.
  B Process–Process: analysis – testing – changes
  C Product–Product: hardware – system
  E Process–Product: analysis, testing, changes – hardware
- *Update* structural *analysis* following *design* changes to find problems earlier.
  B Process–Process: update – analysis – design
- Avoid assessing *design changes* from a narrow, discipline–oriented *view*.
  B Process–Process: design – changes – view

71 Make Sure Ground Support Equipment Cannot Damage Flight Hardware

- Ensure heritage *thermostats* and *relays* properly function when the *system* is *redesign*ed for higher voltages.
  C Product–Product: thermostats – relays – system
  E Product–Process: system – redesign

- Provide ample *test* instrumentation to validate that all *components* of a *system* are functioning properly, and always check for unplanned current draw.
  E Process–Product: test – components
  C Product–Product: components – system
- Individual heater *circuits* should not draw more than two amps to prevent *thermostats* from being damaged by self heating (each of the Apollo 13 switches drew six amps).
  C Product–Product: circuits – thermostats
- Thoroughly *test subsystems* that are not exercised until they are integrated into the main *spacecraft* (such as propulsion lines) during system thermal vacuum test.
  E Process–Product: test – subsystems
  C Product–Product: subsystems – spacecraft

72 Prevent Failures in Support Equipment from Propagating into Flight Boxes

- Buffer test point outputs so shorts in *test* will not damage *flight hardware*.
  E Process–Product: test – hardware
  B Process–Process: test – flight
- Implement abort logic in automated *test equipment* to prevent damage if a failure occurs.
  E Process–Product: test – equipment
- Thoroughly *understand* the inner *workings* of any *item* that interacts with *flight hardware*.
  G People - Process - Product: understand – workings – hardware
  E Process–Product: flight – hardware
  C Product–Product: item – hardware
- Reed *relays*, commonly used in control *circuits*, consist of two overlapping iron *strips* enclosed in a glass *tube*. The *contacts* are readily closed with a magnetic field applied via the surrounding *coils*. The strips should spring back to their normally open position after the field is turned off, but residual magnetism or *magnet*ic contaminants sometimes keep them stuck closed.
  C Product–Product: relays – circuits – strips – tube – contacts – coils – magnet

73 Trace All Software Changes Back to System Requirements and Specifications–Do Not Simply Modify the Code

- Any *software* that commands a *satellite* is *mission* critical, even though it may not be embedded in the *flight vehicle*.
  C Product–Product: software – satellite – vehicle
  B Process–Process: mission – flight
  E Process–Product: flight – vehicle
- *Validate changes* in *mission*–critical *software* with more vigor than the original *development*. Rigorous *formal testing* is essential.
  B Process–Process: validate – changes – mission – development
  E Process–Product: changes – software; testing – formal*
- Always *specify* the *units* in *requirements* and interface *specifications*.
  E Process–Product: specify – units
  C Product–Product: units – requirements* – specifications*
- Generate expected results used in verification *tests* independently, in accordance with *system requirements*.
  E Process–Product: tests – requirements*
  C Product–Product: system – requirements*

74 Understand Why Warning Lights Come On Before Disabling Them

- Operate environmental *tests* with the same degree of care as space *operation*.
  B Process–Process: test – operation
- *Develop test* contingency *plan*s and failure–mode–and–effect–analyses for ground *support equipment* (for example, analyze the likelihood of contamination in case the thermal vacuum facility loses power).
  B Process–Process: develop – test – plan – support
  E Process–Product: support – equipment
- If turning off a *piece* of *test equipment* can endanger *flight hardware*, such equipment must not be allowed to shut down autonomously.
  E Process–Product: test, flight – equipment, hardware
  B Process–Process: test – flight
  C Product–Product: equipment – hardware

75 Protect High–Voltage Equipment from Contamination

- *Design* high–voltage *equipment* to withstand *mishandling*.
  B Process–Process: design – mishandling
  E Process–Product: design – equipment
- Properly vent enclosed *storage* areas to eliminate corona and arcing caused by out–gassing and pressure *buildup*.
  B Process–Process: storage – buildup

- Thoroughly *test* the entire *circuit* if a high voltage is expected.
  E Process–Product: test – circuit

76 Make Sure Someone Takes Responsibility for Each Interface

- *Check* ground operation *procedures* and support *equipment* to avoid damage to *flight hardware*.
  E Process–Product: check – procedures*
  C Product–Product: procedures – equipment – hardware
  B Process–Process: check – flight
- Ensure *interfaces* between two *organizations* are worked out in detail, *agree*d to by both sides, and *document*ed.
  G People – Process - Product: organizations – agree – interfaces
  B Process–Process: agree – document
- *Bound* each *requirement* within a range.
  E Process–Product: bound – requirement*
- The Importance of Stating *TBDs*. *Agency* B's cooling *plan* stated that the *equipment* would be set to "agency A value" or "desired" flow rate." The two *partners review*ed the plan step by step, never realizing that this number had not been agreed upon.
  G People – Process - Product: partners – review – TBDs*
  D People–Process: agency – plan
  B Process–Process: plan – review
  A People–People: agency – partners
  C Product–Product: TBDs – equipment
- *Stating* "set to *TBD* ± TBD units (agency A value to be supplied)" would have raised a flag and avoided the mis*understanding*.
  G People – Process - Product: understanding – stating – TBD*

77 Make Sure Sequential Safety Devices Operate Independently

- Beware that many programmable *devices* do not follow their truth *tables* at power–on.
  C Product–Product: devices – tables*
- After the *bus* power is switched to the pyro *box* via a *relay*, the controller (a field programmable gate *array*, FPGA) should be safe and initialized at the direction of an oscillator clock.
  C Product–Product: bus – box – relay – array

78 Thermal Blankets and Tie–down Cables Can Jam Mechanisms

- *Anticipate* the errant movement and expansion of flexible *materials*, such as *wires* and *blankets*.
  C Product–Product: materials – wires – blankets
  E Process–Product: anticipate – materials
- *Allow* thermal *blankets* to vent whenever possible.
  E Process–Product: allow – blankets
- *Avoid* protrusions or sharp *edges* that can snag soft *items*.
  E Process–Product: avoid – edges*
  C Product–Product: edges* – items
- Indicate the presence of soft *goods* on top–level *assembly drawings* to draw attention to the risks of interference and obstruction problems.
  E Process–Product: assembly – goods
  C Product–Product: goods – drawings

79 Make Sure Software and Hardware Engineers Communicate with Each Other

- *Make sure* no single *parameter* error or single *spacecraft* malfunction can cause endless *cycling* (for example, by enabling the watchdog function to switch to a recovery mode after a few "try agains").
  E Process–Product: make sure – parameter*
  C Product–Product: parameter* – spacecraft
  B Process–Process: make sure – cycling
- Double–*check* last–minute *code changes*.
  B Process–Process: check – changes
  E Process–Product: changes – code
- Problems in embedded *systems* are not always due to random *hardware* defects. Pause and *think* before *inflicting* the same *software* flaw on the redundant side.
  G People – Process - Product: think – inflicting – software
  C Product–Product: systems – hardware – software
- The *computer* uses an independently *clock*ed watchdog function to enable switching to the redundant *CPU* if the primary side malfunctions (for example, due to radiation damage).
  C Product–Product: computer – clock – CPU
- The final *software* mistakenly set the watchdog *counter* to 0.1–s, but it took the *hardware* about a third of a second to boot.

The *CPU* could not finish booting before being reset, and was stuck in an endless loop.

C Product–Product: software – counter – hardware – CPU

80 Check, Double–check, and Triple–check Torquer Phases

- Don't *overlook* simple *tests* that can discover *problems* early.
  G People – Process - Product: overlook – test – problems
- Whenever possible, conduct *independent analyses*.
  D People–Process: independent – analyses
- *Document attitude* control coordinate frames early in *development* to avoid mistakes.
  E Process–Product: document – attitude*
  B Process–Process: document – development
- The calculated moments of inertia, which should have been referenced against the center of gravity, were instead referenced against the origin point on the *drawing*. The mistake was caught by an independent *analysis*.
  E Process–Product: analysis – drawing
- The star *tracker* misbehaved on–orbit because the *vendor alter*ed its coordinate convention but the change notice was not *heeded*.
  G People – Process - Product: vendor – alter – tracker
  B Process–Process: alter – heeded

81 Designate a Responsible Engineer for Complex Equipment

- *Designers* should *inspect* actual *hardware*.
  G People-Process - Product: designers – inspect – hardware
- *Analysis* does not obviate the need to *test*.
  B Process–Process: analysis – test
- Supersonic air rammed through a supposedly sealed *tunnel* on the shield, generating excessive lift that broke the *shield* as well as a nearby *solar array*.
  C Product–Product: tunnel – shield – solar array

82 Understand Transient Behavior of Analog Circuits

- *Check* time–dependent *circuit* behavior, and bound transients in *specifications*.
  E Process–Product: check – circuit
- Do not qualify a *design* solely because a *unit* worked. Measure *circuit* parameters and *verify* that positive margins exist.
  E Process–Product: design – unit; verify – circuit
  B Process–Process: design – verify
  C Product–Product: unit – circuit
- *Analyze* instrumentation *data*, which can provide more engineering information such as post–fire conduction (which may drain flight battery).
  E Process–Product: analyze – data
- *Understand* how *circuits* are typically *design*ed and *test*ed before inventing novel approaches.
  G People – Process - Product: understand – design – circuits
- *Qualify* pyro *devices* by conducting lot acceptance testing.
  E Process–Product: qualify – devices
- *Review* the Pyroinitiator User's *Guide* published by NASA (JSC–28596A).
  E Process–Product: review – guide

83 Put Critical Analyses Under Configuration Control

- Do not *assume* the first, easiest *explanation* is the correct one.
  B Process–Process: assume – explanation
- Refrain from using check–valves as sole means for isolation, as they can chatter or leak (the check–*valve design* and *assembly* process on this *launcher* was particularly prone to seize in the open position). See Check–Valve Reliability in Aerospace Applications, NASA Preferred Reliability Practice No. PD–ED–1267, for additional information.
  E Process–Product: design – valves
  B Process–Process: design – assembly
  C Product–Product: valve – launcher
- The failure cause was found in out–of–family *data* from successful *flights* between the two failures. Notice that a process *change*, chosen to reduce *development costing*, chilled the engine so much that ingressing air could freeze.
  E Process–Product: flights – data
  B Process–Process: flights – change – development – costing

84 Check Start–up Circuit Behavior, Particularly at Low Temperatures

- Use fault–tolerance *circuits* to protect upstream *assets*, not load *units*. Better yet, use dual–level current limiters to protect load units during ground *tests*. But for *flight*, protect only the source circuits.

C Product–Product: circuits – assets – units
E Process–Product: tests – circuits
B Process–Process: tests – flight

- *Redesign* fault–tolerance *circuits* when the load *units* have been substantially *alter*ed.
  B Process–Process: redesign – alter
  E Process–Product: redesign – circuits
  C Product–Product: circuits – units
- The multiplex *chips* draw 0.25 mA during *operation*, but as much as 5 mA during cold power up.
  E Product–Process: chips – operation
- When the current draw exceeds the power source's capability, the *unit* would continue trying to *reboot*. The primary *computer* timed out; its back–up finally succeeded in booting after the *chips* warmed up.
  E Process–Product: reboot – unit
  C Product–Product: unit – computer – chips

85 Systems and Software Engineering Should Actively Coordinate

- *Test* the specific *configuration* that will be *flown*.
  B Process–Process: test – configuration – flown
- Conduct *tests* and *reviews* to validate that the *requirements* are met, rather than that the *drawings* are correctly implemented.
  E Process–Product: test, review – requirements*, drawings
  B Process–Process: test – review
  C Product–Product: requirements* – drawings
- Actively involve systems *engineers* in *software development* activities, and formally control all *system* (including software) interfaces.
  G People – Process - Product: engineers – development – software
  C Product–Product: software – system

86 Hand–Over Logic Tree Must Be Unambiguous

- Conduct redundancy switching *analysis* to ensure a fail–safe transfer between multiple, or redundant, *controllers*. Postulate all credible failure paths (such as part failure, start–up transients, latch–up, overvoltage, and EMI) and determine the effect on the *switch*ing process. *Make sure* glitches in one *unit* will not propagate across *interfaces*.
  B Process–Process: analysis – make sure
  E Process–Product: analysis – controllers; make sure – unit, interfaces
  C Product–Product: controllers – switch – unit – interfaces
- Guard against radio frequency (RF) *interference* from multiple *sources*.
  C Product–Product: interference* – sources
- A *study* of *missiles* converted for suborbital or space *launches* found that the largest cause of failure was electromagnetic interference (*EMI*).
  B Process–Process: study – launches
  E Process–Product: launches – missiles
  C Product–Product: missiles – EMI*

87 Avoid Repeating Other People's Mistakes

- *Study* past failures that involved similar *technologies* and implement appropriate corrective actions.
  E Process–Product: study – technologies
- Ensure *subcontractors* discuss relevant lessons with the *prime contractor*.
  A People–People: subcontractor – prime contractor
- As a *rocket* ascends, decreasing atmospheric pressure causes its flame to spread out. The *designers* of this failed launcher conducted static firings, but did not run sufficient computational fluid dynamics *modeling*. Thus, they did not anticipate the *conflagration* or the need to protect the *cable*.
  G People – Process - Product: designers – modeling – rocket
  C Product–Product: rocket – cable
  B Process–Process: modeling – configuration
  E Process–Product: configuration – cable

88 Verify Each Operation Step

- Implement a discrete *verification* step for each critical *task*.
  B Process–Process: verification – task
- Require positive *confirm*ation before hazardous *commands* can be acted upon.
  B Process–Process: confirm – commands
- Do not *deviate* from written *procedures*.
  E Process–Product: deviate – procedures*
- *Handle* space *hardware* carefully.

E Process–Product: handle – hardware

- As a thunderstorm approached a launch pad, *workers draped* a rain shield over a *satellite* being processed in the White Room. The *shield* consisted of overlapping strips of waterproof *cloth*, secured with adhesive *tapes*. The *installation* instructions stated, "ensure both top and bottom sides of seam are *tape*d." Nonetheless, the lower *side* was neglected, nor was there verification.
  G People – Process - Product: workers – draped – satellite
  C Product–Product: satellite – side – shield – cloth – tapes
  B Process–Process: installation – verification
  E Process–Product: installation – tape; verification – side
- Rainwater poured through the building's leaks. The weak rain *shield* collapsed, drenching the *satellite*. *Launch* had to be *delay*ed for years.
  C Product–Product: shield – satellite
  E Process–Product: launch – satellite
  B Process–Process: launch – delay

89 Prevent Hardware Fratricide

- *Ensure* the neighboring *units* survive after the primary *device* operates.
  E Process–Product: ensure – units, device
  C Product–Product: units – device
- *Qualify* ordnance *devices* in their *operation*al environment.
  B Process–Process: qualify – operation
  E Process–Product: qualify – devices
- A *review* of a previous *mission* revealed that several non–critical *pins* had disengaged. Unfortunately, these warning *signs* were not heeded and the *connectors* were not *redesigned*.
  B Process–Process: review – mission – redesigned
  E Process–Product: review – pins, signs, connectors
  C Product–Product: pins – signs – connectors
- A *launcher* used shaped *charges* to separate the stages. The *initiator* on one end fired first, disabling the other end of the charge and preventing the *structure* underneath the damaged initiator from tearing apart. The *vehicle* jackknifed.
  C Product–Product: launcher – chargers – initiator – structure – vehicle

90 Account for All Loose Materials

- Make sure loose, non–serialized *materials* (such as wipe cloth) used during *assembly* are carefully *accounted* for.
  E Process–Product: assembly – materials
  B Process–Process: assembly – accounted
- *Correct* the *root cause* of in–Process anomalies.
  B Process–Process: correct – root cause
- *Keep* accurate *record*s of all "non-flight" *installations*.
  B Process–Process: keep – record – installations
- Take *photos* frequently during *assembly*.
  B Process–Process: photo – assembly
- *Design hardware* to minimize areas that cannot be easily *inspected*, and avoid the use of potential *contaminants* whenever possible.
  E Process–Product: design – hardware
  C Product–Product: hardware – contaminants*
  B Process–Process: design – inspected
- Keep *hardware closed* when access is not needed.
  E Product–Process: hardware – closed
- *Review* out–of–flow processes to ensure no steps are *bypass*ed.
  B Process–Process: review – bypass
- Debris contamination spoiled five foreign *launches* between 1990 and 1999, including several caused by *rags* clogging propulsion *lines*.
  E Process–Product: launches – rags
  C Product–Product: rags – lines
- Debris such as paper clips left in RF cavities repeatedly caused *test* failures on a *satellite* program. The *contractor* finally *develop*ed an electromagnetic *probe* to sweep all cavities before they were sealed.
  E Process–Product: test – satellite
  G People – Process - Product: contractor – develop – probe
  B Process–Process: develop – test
- A jet engine *contractor* suffered several failures caused by *bolts* or *tools* being left inside *test units*. The management subsequently required an *inspector* to go inside the inlet to *check* for debris using a *flashlight*. Right after the new procedure was implemented, the *engine* blew up. The flashlight was left behind. (From "Augustine's Laws.")
  G People – Process - Product: contractor – test – units; inspector – check – flashlight

C Product–Product: bolts – tools – units

91 Ensure Critical Systems Are Tolerant of Transient Power Loss

- *Ensure* the onboard *computer* retains "most recent state" information so that if a glitch causes the loss of "present state" *data*, the vehicle can revert to a survivable *configuration*.
  E Process–Product: ensure – computer
  B Process–Process: ensure – configuration
  C Product–Product: computer – data
- Anticipate *wiring* problems, and provide redundant power *sources* to critical *systems*, including lock–in power *circuits* to prevent *hardware* reset.
  C Product–Product: wiring – sources – systems – circuits – hardware
- *Recognize* the need to address *weaknesses* in non-propulsive *systems*.
  E Process–Product: recognize – weaknesses*
  C Product–Product: weaknesses* – systems
- After this incident, the *contractor redesigned* the 30–year old control *electronics* to provide redundant power and guidance. A *sister* launch vehicle program, however, did not make a similar *change*.
  G People – Process - Product: contractor – redesigned – electronics
  D People–Process: sister – change
- Years later, the second program suffered a failure. Apparently, a defective power *cable* shorted intermittently, causing the guidance *computer* to reset and the inertial measurement *unit* to lose reference.
  C Product–Product: cable – computer – unit
- The *launcher* had miles of *wires*. Forty–four *repairs* had been made on this particular *vehicle* alone. In retrospect, it was clearly impossible to *inspect* out every wiring defect, and the decision not to provide redundant power proved *costly*.
  C Product–Product: launcher – wires – vehicle
  E Process–Product: repairs – vehicle
  B Process–Process: repairs – inspect – costly
- *Cabling* defects led to the most costly unmanned *launch* failure
  E Process–Product: launch – cabling

92 Rigorously Determine the Root Causes of Test Failures

- New *technologies* require rigorous *qualification*, *analysis* of *design changes*, and a thorough understanding of failure modes.
  E Product–Process: technologies – qualification, analysis, design, changes
  B Process–Process: qualification – analysis – design – changes
- *Audit* a *vendor*'s *manufacturing* process, conduct destructive physical *analysis* of sample *parts*, and ascertain the root causes of all anomalies.
  G People – Process - Product: vendor – manufacturing – parts
  B Process–Process: manufacturing – analysis – audit
  E Process–Product: analysis – parts
- *Review* the *materials* and processes for each new application *drawing*.
  E Process–Product: review – materials
  C Product–Product: materials – drawings
- *Guard* against known *materials* incompatibilities (gold/tin intermetallics can embrittle solder joints, for example).
  E Process–Product: guard – materials

93 Always Ascertain the Direction of Current Flow

- *Make sure* that *engineers* understand how the *system* or *component* should function during *test*.
  G People – Process - Product: engineers – make sure – system, component
  B Process–Process: make sure – test
  C Product–Product: system – component
  E Process–Product: test – system
- Thoroughly *verify* interfaces of *subcontract*ed *items*, particularly when the *suppliers* use different engineering *conventions*.
  G People – Process - Product: subcontract – verify – items
  A People–People: subcontract – suppliers
  F People–Product: suppliers – conventions*
- Use an engineering *model* to *verify* interfaces early.
  B Process–Process: model – verify
- A *preflight check* found two *hardware modules* wired in the opposite *polarity*. Both *subcontractors* reversed their *cables*. The *launch* failed.
  E Process–Product: preflight – hardware
  C Product–Product: hardware – modules – polarity*
  G People – Process - Product: subcontractors – cable – launch

94 Provide Debug Features in Flight Software to Assist Anomaly Resolution

- Ensure that commercial *software*, especially the operating *system*, allows access to internal *information* and is compatible with *development* debug tools.
  C Product–Product: software, system, information
  E Process–Product: development – software
- *Test* for off–nominal *conditions*, both "better" and "worse" than expected (for example, at higher throughput rate), to see if the *system* misbehaves.
  E Process–Product: test – conditions*
  C Product–Product: system – conditions*
- Leave *debug* capabilities embedded in the operational *system*.
  E Process–Product: debug – system
- Shared *functions* must be thoroughly *tested*, especially for timing.
  B Process–Process: tested – functions
- Because the *bus* and *instruments* share the *processor*, job allocation is vital. The highest priority is given to *data* management, followed by bus *tasks* and by science *activities*. If data management tasks cannot complete within the watchdog's 125 millisecond cycle, an anomaly is assumed and the *computer* is reset.
  C Product–Product: instruments – processor – data – computer
- *Data* from the *bus* and *payloads* flow through a 1553 data bus, but one *instrument* is processed directly.
  C Product–Product: data – payload – instrument
- That *sensor* shares a *software* function with the transaction manager–not a prudent *design* but normally not a problem.
  C Product–Product: sensor – software
  E Process–Product: design – software
- Turning on "priority inheritance" *options* for that particular *thread* solves this problem. This option is not normally used as default due to *performance* concerns.
  B Process–Process: options – threads
  E Process–Product: options – performance*

95 Ensure Heritage Designs Can Operate in the New Application Environment

- Avoid relying on short–term *tests* (days to months) to *confirm* long–term *reliability*.
  B Process–Process: test – confirm – reliability
- *Audit vendor material* lists to ensure completeness.
  G People – Process - Product: vendor – audit – material
- *Account* for vapor diffusion in propulsion *subsystem design*.
  B Process–Process: account – design
  E Process–Product: design – subsystem

96 Tests Must Independently Verify Development Results

- Use simple *tools* to *crosscheck* elaborate *tests*.
  E Process–Product: crosscheck – tools
- Scrutinize *test equipment*, *analysis*, or *algorithms reuse*d from *design* or *manufacturing* for possible single–Point failure.
  E Process–Product: test – equipment, algorithms
  B Process–Process: analysis – reuse – design – manufacturing
- Missing coating near the *cap* aperture caused the *operator* to *aim* the light at the cap instead of at the *rod*.
  G People – Process - Product: operator – aim – cap
  C Product–Product: cap – rod
- The mis–focusing prevented the metering rod from reaching the *lens*, but the *technicians* simply *extended* the *rod* by inserting a few *washers*.
  G People – Process - Product: technicians – extended – lens
  C Product–Product: lens – rod – washers
- That in itself should have alerted *people*, because clearly there should not be a *need* for any unexpected *washers* to be added, said the *investigation board*.
  G People – Process - Product: people – need – washers
  A People–People: people – board
  B Process–Process: need – investigation

97 Control Hardware and Software Configurations Before, During, and After Tests

- Always *ascertain* G&C *actuator* phasing.
  E Process–Product: ascertain – actuator
- Ensure domain *engineers own* all aspects of their *subsystems*.
  G People – Process - Product: engineers – own – subsystems
- Conduct end–to–end *test*ing in the *flight configuration*.
  B Process–Process: test – flight – configuration

- Take plenty of *photographs* during *assembly*.
  B Process–Process: photography – assembly
- *Document* G&C *subsystem*–level alignment. See *Guideline* GD–ED–2211 from NASA Technical Memorandum 4322A, for example.
  E Process–Product: document – subsystem
  C Product–Product: subsystem – guideline*

98 Guard Against Post–Firing Conduction of Pyro Initiators

- *Protect* firing *circuits* against sneak currents and line–to–ground shorts.
  E Process–Product: protect – circuits
- *Components* such as step *motors* and pyro *circuits* that experience sudden current changes should be isolated from all other current–carrying circuits including *electrical* power, electrical control, RF transmission *lines*, and monitoring circuitry. For additional information, see Electromagnetic Interference Analysis of Circuit Transients, NASA Preferred Reliability Practice No. PD–AP–1308, for example.
  C Product–Product: components – motors – circuits – electrical – lines
- *Check circuit designs* against Electroexplosive Subsystem Safety Requirements and Test Methods for Space Systems (MIL–STD–1576), NASA Standard Initiator User's Guide (JSC–28596A), and Electrical Grounding Architecture for Unmanned Spacecraft (NASA–HDBK–4001).
  E Process–Product: check – circuit
  B Process–Process: check – design
- Post–fire *plasma* shorts can drain *batteries*. See Journal of Spacecraft and Rockets, 36, 586–590 (1999).
  C Product–Product: plasma – batteries
- Drive *elements* can be disabled by residual current, and should be inspected after ground live *tests*. In one case, an *inspection* found a damaged fusing *resistor*, which would have prevented *in–flight* firing. Between 3% and 5% of firings result in. conduction.
  C Product–Product: elements – resistor
  E Process–Product: inspection – resistor
  B Process–Process: inspection – in–flight

99 Have the Model's Originator Check the Analysis

- Double *check* all *analysis models*, assumptions, methods, and predictions.
  B Process–Process: check – analysis – models
- *Develop* a rigorous process for using *experience* as a basis for accepting further *designs* and *equipment.*
  G People – Process - Product: experience – develop – equipment
  B Process–Process: develop – design
- Have the original *analyst review* final *product*.
  G People – Process - Product: analyst – review – product
- Make sure key *subcontractors accept* how their *product* is being used.
  G People – Process - Product: subcontractor – accept – product

100 Make Sure Safety Mechanisms Are Truly Independent

- Ensure safing *mechanisms* will prevent one *design* error from causing a cascade of irreversible failures. In this case, one error could have activated all the *heaters*, and the *solar arrays* might have been *deployed* prematurely.
  C Product–Product: mechanisms* – heartes – solar arrays
  E Process–Product: design – mechanisms*
  B Process–Process: design – deployed
- *Check* for failure *mechanisms* during extended *operation* even if that is not the intended application. If prolonged operation leads to catastrophic failure, provide *circuit* interrupts, time–out protection, or a graceful degradation mechanism.
  E Process–Product: check – mechanisms*
  B Process–Process: check – operation
  C Product–Product: mechanisms* – circuit
- *Review* special *design requirements* for *FPGAs*.
  B Process–Process: review – design
  E Process–Product: design – requirements*
  C Product–Product: requirements* – FPGAs

Appendix B: SI Element Coding Sheets

Table 11. *People* SI Element Coding Sheet

Original Concept	PR # - LL #	Coded Concept
Aerospace	9-2, 20-4	stakeholder
agency	76-4	developer
analysts	26-4, 99-3	developer
approved	25-2	acquirer
assemblers	65-3	developer
audience	59-4	acquirer
authorities	40-2	stakeholder
board	96-5	acquirer
builders	4-2	developer
buyers	5-2	acquirer
contractor(s)	5-2, 11-8, 14-3, 23-3, 33-3, 37-2, 87-2, 11-4, 11-6, 14-3, 20-5, 90-9, 90-10, 91-4	developer
controllers	67-1	operator
cross-program	50-2	developer
decision	23-5, 23-4	acquirer
designers	23-6, 26-1, 87-3, 81-1	developer
doubt	38-3	developer
engineer(s)	12-2, 17-1, 49-2, 2-1, 4-1, 12-3, 16-1, 17-1, 49-4, 85-3, 93-1, 97-2	developer
experiences	26-5, 99-2	developer
human error	3-5, 32-3, 43-4, 51-2, 51-3	developer
independent	3-3, 80-2	stakeholder
inspector	90-10	developer
knowledge	50-3	developer
knowledge	67-2	operator
mistakes	34-2	operator
MRB	6-2	acquirer
NASA	21-5, 55-4	stakeholder
one	3-2	developer
organization(s)	2-4, 50-2, 11-3, 2-6, 76-2	developer
other	2-4	developer
overall	3-3	acquirer
overlook	80-1	developer
ownership	3-3	operator
partners	76-4	supplier
people	96-5	developer
personnel	23-5, 64-2, 29-1, 47-1, 54-1	developer
personnel	17-5	operator
planners	5-3	developer
program	67-3	developer
program manager	5-4	acquirer
program office	11-8, 37-2	acquirer
program(s)	26-4, 23-3, 2-2	acquirer
programmer	43-5	developer
public	40-2	stakeholder
resources	23-5	developer
satellite operators	17-1	operator
side	11-3	developer
sister	91-4	developer
space business	3-5	acquirer
staff	29-2	developer
subcontractor(s)	5-4, 87-2, 93-2, 93-4, 99-4	developer
system integrator	11-8	developer
team	12-2, 61-3	developer
technician(s)	4-5, 96-4	developer

Original Concept	PR # - LL #	Coded Concept
teleconference	59-4	developer
think	79-3	developer
understand	38-2, 40-1, 82-4, 76-5, 65-2, 72-3	developer
vendor	54-6, 80-5, 95-2, 92-2	supplier
vigilant	52-3	developer
workers	88-5	developer

Table 12. *Process* SI Element Coding Sheet

Original Concept	PR # - LL #	Coded Concept
accept - any variation	7-3, 99-4	evaluate
account	95-3, 90-1	manage
add	30-2	design
agree	76-2	manage
aim	96-3	manage
allow	63-3, 28-2	design
alter - any variation	80-5, 45-2, 84-2, 80-5	implement
analysis - any variation	18-2, 2-3, 2-2, 20-4, 26-1, 48-3, 70-1, 81-2, 86-1, 96-2, 16-2, 99-1, 47-2, 92-2, 26-2, 92-1, 66-2, 70-2, 13-5, 59-2, 38-4, 13-2, 11-8, 16-2, 56-2, 70-1, 80-4, 86-1, 92-2, 48-2, 62-1, 82-3, 41-2, 13-5, 59-2, 44-4, 21-4, 92-1, 20-4	define
anomalies	65-1	deploy
anticipate	28-1	manage
applied	43-5	design
archive	57-3	manage
ascertain	56-3, 53-2, 97-1	evaluate
assemble - any variation	47-1, 61-2, 90-1, 83-2, 90-4, 97-4, 63-2, 61-2, 78-4, 90-1, 47-1	implement
assume	83-1	define
audit	95-2, 92-2	evaluate
automatic	47-3, 43-4	design
avail	23-5	manage
avoid	15-1, 19-5, 32-2, 17-6, 78-3	manage
back-up	39-3	deploy
baked	51-7	evaluate
balanced	27-1	design
bar coding	51-1	manage
bound	76-3	define
braze	15-1	implement
budget - link	16-2, 46-1	design
build - any variation	64-1, 15-2, 4-5, 75-2	implement
bypass	90-7	design
came from	51-5	define
chamfering	69-4	implement
change - any variation	25-2, 25-3, 83-3, 48-3, 38-4, 39-6, 33-4, 70-3, 48-1, 73-2, 79-2, 70-1, 28-3, 92-1, 25-3, 30-3, 73-2, 79-2, 70-1, 45-1, 22-4	implement
check	15-3, 38-4, 61-1, 66-4, 76-1, 79-2, 98-3, 99-1, 100-2, 39-6, 15-2, 43-3, 60-4, 64-6, 66-4, 76-1, 82-1, 98-3, 100-2, 96-1, 24-1, 15-3, 90-10, 24-5	evaluate
clean - any variation	21-2, 45-5, 14-5, 45-4	deploy
closed	90-6	deploy
coating - any variation	5-5, 47-5	deploy
collect - any variation	46-4, 46-3	manage
combine	38-5	design
commands	88-2	deploy
communicate	33-3	manage
conduct	69-1	manage
configure - any variation	39-5, 91-1, 60-2, 87-3, 7-2, 42-4, 85-1, 97-3, 39-5, 87-3, 53-1, 16-3, 7-2	manage
confirm	88-2, 95-1	evaluate
consider	59-1, 65-1	manage

Original Concept	PR # - LL #	Coded Concept
control	16-1, 34-3	manage
correct - any variation	64-1, 44-1, 34-2, 90-2	implement
corrosion	45-5	deploy
cost - any variation	83-3, 37-4, 91-6, 23-4, 2-6	manage
cycling	79-1	evaluate
damage	28-4, 41-3, 17-3, 49-3	deploy
debug	94-3	evaluate
delay	88-6	deploy
delivery	37-4	manage
deploy - any variation	100-1, 53-4, 63-1, 20-4, 9-4	deploy
design - any variation	71-1, 91-4, 84-2, 89-3	design
detect	28-5	evaluate
determine	4-1, 44-1	define
develop - any variation	3-1, 13-5, 59-2, 74-2, 90-9, 99-2, 25-2, 1-3, 60-2, 80-3, 83-3, 16-1, 73-2, 94-1, 68-1, 23-1, 3-1, 85-3	implement
deviate - any variation	88-3, 60-1	implement
diagnostic	31-2	evaluate
disassembly	54-5	implement
discover	37-3	define
display	59-3	deploy
document	76-2, 22-1, 32-4, 80-3, 97-5, 26-4	manage
draped	88-5	deploy
effects	45-2	deploy
emission	40-4, 40-3	deploy
ensure	91-1, 89-1	evaluate
epoxy	28-4	implement
estimation	11-2	define
evaluate - any variation	19-3, 41-1, 39-1	evaluate
explain	83-1, 39-1	define
extended	96-4	implement
fabrication	31-1	implement
facilities	54-1	deploy
fail-safe	32-3	design
failure	43-5	deploy
field	22-1	deploy
firing	14-2	evaluate
fix	12-6	implement
flexibility	50-1	design
flight - any variation	76-1, 39-5, 33-4, 83-3, 15-4, 73-1, 44-6, 59-4, 69-2, 14-1, 72-1, 74-3, 97-3, 84-1, 11-4, 11-7, 72-3, 73-1, 65-4, 4-4, 4-5, 93-4, 98-5	deploy
flow - any variation	12-4, 5-2, 12-2, 85-1	define
follow	54-1	deploy
functions	94-1	design
fund - any variation	11-8	manage
guard	92-4	deploy
handle - any variation	30-1, 88-4, 5-3, 9-2, 28-1, 35-2, 41-3, 57-2, 75-1	deploy
health	29-3	deploy
heeded	80-5	deploy
housekeeping	46-5	deploy
impacts	42-2	deploy
implementation	26-1	implement
increments	35-4	manage
inflicting	79-3	design
initialization	61-1	deploy
inspect - any variation	61-2, 66-4, 47-2, 54-5, 90-5, 5-2, 19-2, 98-5, 91-6, 66-4, 47-4, 98-5, 19-2, 28-2, 26-1, 81-1	evaluate
install - any variation	64-5, 88-5, 64-3, 61-1, 90-3, 56-5. 88-5	deploy
investigate	96-5, 55-4	define
keep	90-3	deploy

Original Concept	PR # - LL #	Coded Concept
label	68-2	manage
launch - any variation	44-6, 63-1, 59-4, 88-6, 15-3, 14-5, 57-5, 86-3, 15-2, 59-5, 62-3, 62-4, 66-1, 67-1, 88-6, 91-7, 86-3, 90-8, 1-4	deploy
lifecycle	37-1	manage
looked	51-5	define
maintain - any variation	15-4, 54-5, 24-1, 37-2	deploy
make sure	86-1, 79-1, 93-1	evaluate
manage - any variation	47-3, 60-2	manage
maneuvers	27-4	deploy
manufacture - any variation	92-2, 96-2, 57-3, 22-1, 41-3, 6-1, 55-3	implement
mechanism	54-2, 44-3	implement
migration	27-2	deploy
mission	67-4, 25-1, 65-2, 73-1, 2-1, 89-3, 33-2, 65-5, 73-2	deploy
mitigate	13-3, 36-3	manage
mixed-up	51-4	deploy
mode	38-1	deploy
model - any variation	61-3, 67-3, 2-1, 2-2, 2-5, 7-2, 14-1, 26-2, 26-3, 36-2, 37-3, 43-2, 52-1, 87-3, 93-3, 99-1	evaluate
modification	41-1	implement
need	96-5	deploy
observe	67-2	define
operation - any variation	3-1, 1-2, 9-1, 89-2, 18-5, 100-2, 21-1, 74-1, 38-1, 23-2, 63-1, 54-6, 29-4, 11-1, 57-1, 63-2, 39-6, 23-2, 29-4, 53-3, 25-4, 18-3, 84-3	deploy
options	94-8	manage
orienting	17-3	deploy
overlook	51-3	define
own	97-2	evaluate
packaging	47-2	deploy
photo - any variation	90-4, 97-4	implement
place	27-4	design
plan - any variation	76-4, 56-1, 74-2, 60-3, 17-2, 40-2	manage
prevent	54-2	design
production	51-1	implement
profiles	31-2	design
protect	21-2, 18-1	deploy
pumped	51-7	evaluate
pursue	65-6	design
qualify - any variation	7-3, 92-1, 89-2, 14-2, 4-3, 28-3, 82-5, 89-2, 92-1, 3-1, 18-7, 1-3, 4-3	evaluate
reboot - any variation	84-4, 17-4	deploy
receive	67-1	deploy
recognize	91-3	define
record	90-3, 64-3	manage
reduce	11-4	design
redundancy	18-1, 54-4	design
re-examine	18-4	evaluate
reliability - any variation	96-3, 18-2, 18-7, 47-1, 95-1, 46-3	design
removal	57-2	deploy
repair - any variation	15-2, 6-1, 32-5, 91-6, 6-3	implement
replace	14-2, 14-3	implement
reprogrammability	50-1	design
rerun	12-6	evaluate
resolution	2-3	implement
retention	21-6	design
retrofit	57-5	implement
reuse - any variation	96-2, 18-4, 23-3	design
review - any variation	6-2, 12-4, 16-1, 42-2, 54-1, 66-2, 89-3, 90-7, 100-3, 5-4, 16-1, 17-4, 23-6, 76-4, 99-3	evaluate
rework	32-5	implement
root - any variation	34-2, 90-2, 32-4, 55-1	define
safety	49-2, 54-1, 29-3, 5-4, 28-2	design
schedule - any variation	20-5, 46-5, 37-4, 50-5	manage

Original Concept	PR # - LL #	Coded Concept
screened	22-2	evaluate
search	55-1	define
service	54-6	deploy
simplify	63-1	design
simulate - any variation	36-2, 7-2, 14-1, 43-2, 29-1, 7-2	evaluate
solve	46-4	deploy
space - any variation	8-1, 8-2, 10-1, 20-5, 42-1, 42-2, 49-5	deploy
specify	73-3	design
stages	33-2, 65-3	deploy
stating	76-5	manage
stiction	24-6	deploy
stops	65-5	deploy
store - any variation	5-3, 9-1, 9-2, 21-1, 21-6, 21-7, 22-1, 75-2, 88-5, 90-3	deploy
stretching	52-3	manage
study	69-2, 87-1, 86-3	define
suitability	18-4	design
support	2-3, 74-2, 13-1, 74-2	deploy
sustain	38-3	deploy
task	32-2, 32-1, 88-1	implement
test - any variation	23-6, 7-3, 7-2, 12-6, 14-1, 38-1, 39-1, 42-4, 49-2, 49-3, 49-5, 51-4, 60-1, 64-5, 72-1, 74-1, 74-3, 85-1, 85-2, 95-1, 97-3, 94-4, 18-5, 1-3, 45-4, 84-1, 20-4, 46-1, 5-5, 7-1, 11-1, 74-2, 52-1, 54-6, 69-2, 45-3, 81-2, 56-3, 20-1, 90-9, 93-1, 70-1, 3-1, 35-4, 26-3, 9-1, 36-2, 69-1, 28-3, 39-6, 14-2, 25-3, 66-4, 3-2, 7-2, 11-5, 12-4, 12-5, 12-6, 21-3, 24-3, 34-1, 39-1, 42-4, 49-2, 58-1, 58-2, 58-3, 64-5, 71-2, 71-4, 72-1, 72-2, 74-3, 75-3, 85-2, 90-9, 93-1, 94-2, 96-2, 18-5, 35-3, 36-2, 73-4, 84-1, 52-1, 54-6, 70-1, 9-1	evaluate
thread	94-8	design
time	45-3	design
trace	44-2	define
track	65-1, 12-1	manage
train	17-1	deploy
transitions	13-4, 13-5, 38-1	deploy
trend - any variation	19-3, 19-2, 39-1	define
troubleshoot	53-4, 44-2, 46-2	evaluate
turned-off	17-6	deploy
undetect	36-3	deploy
update	70-2	implement
upgrade	57-1	implement
validate - any variation	2-3, 44-3, 29-4, 73-2	evaluate
verify - any variation	48-3, 25-3, 20-3, 42-3, 82-2, 93-3, 48-1, 43-2, 63-2, 53-6, 93-2, 32-1, 56-1, 88-1, 88-5, 32-5, 1-3, 53-6, 63-2, 19-1	evaluate
view	70-3	define
walkthrough	64-2	evaluate
weld	15-1	implement
workings	72-3	design
wrap	65-3	implement

Table 13. *Product* SI Element Coding Sheet

Original Concept	PR # - LL #	Coded Concept
accessible*	64-6	requirement
acoustic*	24-3	requirement
acrylics	51-6	hardware
actuator	97-1	hardware
adhesive	51-6	hardware
aft part	52-4	hardware
airplane	11-4	system
algorithm	13-5, 13-3, 43-2, 96-2, 27-3	software

Original Concept	PR # - LL #	Coded Concept
amplifier	56-4	hardware
anode	41-5	hardware
anomalies	65-1	data
antenna	9-4, 9-3, 42-5	subsystem
appendages	13-4	hardware
area	63-3	data
array	38-5, 40-4, 53-4, 77-2	hardware
assemblies	21-5	hardware
assets	84-1	hardware
attitude*	67-2, 67-3, 80-3	requirement
axes	21-6, 67-2	hardware
baffles	52-4	hardware
base–Plate	64-4	hardware
batteries	22-1, 22-2, 22-3, 62-3, 98-4, 53-4, 62-2, 30-2, 30-3, 62-4	hardware
behavior*	39-6, 52-1	requirement
blankets	51-6, 78-1, 78-2	hardware
bolts	90-10	hardware
boom	56-4	hardware
booster engine	11-4, 32-5	subsystem
box	58-3, 77-2	hardware
brushes	41-4, 41-5	hardware
bus	49-4, 77-2	segment
cable	10-2, 87-3, 91-5, 51-2, 93-4, 11-1, 91-7	hardware
cadmium	49-5	hardware
camera	61-3	hardware
cap	96-3	hardware
capacitor	38-5	hardware
capillary	62-4	hardware
carrier	11-7	system
case	6-3	hardware
catalog items	5-3	data
cells	22-3, 26-6, 62-3	hardware
chamber	4-4, 45-4, 49-5, 69-3	hardware
channels	46-2	hardware
chargers	89-4	hardware
chassis	56-5	hardware
chips	84-3, 84-4	hardware
chloride	45-5	hardware
circuit	82-2, 98-3, 19-5, 19-5, 56-5, 63-3, 100-2, 75-3, 82-1, 82-2, 84-2, 24-2, 10-2, 24-2, 71-3, 72-4, 84-1, 84-2, 91-2, 98-2, 82-4, 84-1, 98-1	hardware
clock	79-4	hardware
cloth	88-5	hardware
code	35-4, 18-6, 79-2	software
coils	53-6, 53-5, 72-4	hardware
commercial parts	5-3	hardware
component	93-1, 19-6, 49-5, 93-1, 71-2, 30-1, 98-2, 44-3	hardware
computer	91-1, 30-3, 39-6, 79-4, 84-4, 91-5, 94-5	hardware
conditions*	9-1, 42-3, 94-2, 44-4,	requirement
connector	62-4, 64-6, 65-3, 63-2, 68-2, 65-5, 89-3, 11-1, 15-2, 11-1, 15-2, 46-5, 89-3	hardware
constant	3-3	requirement
contacts	72-4	hardware
contamination*	90-5, 16-1, 16-4, 16-3	requirement
contingencies*	60-3	requirement
controller	56-4, 86-1, 30-3, 86-1	hardware
conventions*	93-2	requirement
core	53-5	hardware
counter	79-5	hardware
coverage*	53-3	requirement
CPU	79-4, 79-5	hardware

Original Concept	PR # - LL #	Coded Concept
crosslinks	40-3	data
cryogens	54-3	hardware
current	53-5	data
data	24-5, 43-4, 16-2, 29-4, 43-3, 46-3, 46-4, 59-2, 66-4, 19-2, 29-4, 35-2, 43-3, 46-4, 67-3, 94-5, 94-6, 59-4, 19-3, 35-2, 39-1, 59-3, 82-3, 83-3, 16-2, 46-3, 91-1	data
database	43-2, 3-4, 43-1	software
decimal point	3-7	data
device	89-1, 82-5, 15-3, 77-1, 89-2	hardware
dewar	52-4	hardware
diagrams*	68-1	requirement
dielectrics	42-1, 42-1	hardware
dipole	53-5	hardware
display	3-8, 3-7	hardware
documentation*	18-4	requirement
Doppler Shift*	23-6	requirement
drains	57-2	hardware
drawings	80-4, 64-1, 85-2, 64-1, 65-3, 85-2, 78-4, 92-3	requirement
Earth	60-5	data
edges*	78-3	requirement
electrical	68-1, 98-2	hardware
electrode	22-3	hardware
electrolyte	62-4, 22-4, 62-3	hardware
electronics	91-4, 11-6, 11-6, 16-4	hardware
electrostatic*	42-1, 42-5	requirement
elements	98-5	requirement
EMI*	86-3	requirement
engine	45-4	subsystem
equipment	17-6, 99-2, 41-1, 49-1, 74-2, 74-3, 96-2, 13-1, 46-5, 49-2, 52-1, 54-1, 74-3, 76-1, 49-2, 62-1, 72-2, 75-1, 76-4	hardware
ESD*	10-2, 17-6	requirement
excitations	13-3	requirement
facilities*	24-1, 46-5	requirement
failure mode	12-3, 7-3	software
failure paths	19-5	software
filter	38-5, 44-6, 56-4, 56-5, 57-5, 67-3	hardware
firing	14-2	requirement
fittings	15-1, 15-3, 15-1, 15-2, 15-3	hardware
flange	4-5	hardware
flare	15-4	hardware
flashlight	90-10	hardware
fluid lines	15-1	hardware
force*	24-4, 15-4	requirement
formal*	73-2	requirement
formula	43-5	software
forward part	52-4	hardware
FPGAs	100-3	hardware
frequency*	13-2, 13-2, 11-7, 38-5	requirement
friction*	24-6	requirement
fuel	57-2	hardware
fuse	56-5, 19-6	hardware
gas	27-5-C	hardware
gas inlet	32-5-C	hardware
gimbal	50-4-C	hardware
gold	31-4-C	hardware
goods	78-4-C, 78-4	hardware
grain	69-3, 69-4	hardware
ground	39-3	segment
ground equipment	5-1	hardware
grounding*	58-3	requirement

Original Concept	PR # - LL #	Coded Concept
guide	82-6	requirement
guideline*	97-5	requirement
gyros	21-6	hardware
hardware	3-2, 17-3, 23-3, 26-4, 47-1, 72-3, 3-2, 12-6, 16-1, 47-4, 53-1, 57-1, 70-1, 72-1, 72-3, 74-3, 90-5, 93-4, 21-2, 42-2, 49-3, 90-6, 3-2, 16-1, 18-1, 18-6, 21-2, 21-7, 23-3, 36-2, 36-3, 53-1, 54-1, 58-2, 70-1, 76-1, 79-3, 79-5, 90-5, 91-2, 93-4, 81-1, 18-1, 58-1, 61-2, 88-4, 12-6, 72-3, 74-3	hardware
harness	58-2, 11-6, 26-6, 58-2,8-3	hardware
health	39-2	requirement
heater	100-1, 39-6, 30-3, 39-3, 39-4, 39-3, 30-3	hardware
heatsink	47-5	hardware
helium	52-4	hardware
honeycomb cells	1-4	hardware
honeycomb structure	1-1, 1-2	hardware
ICBM	57-5	system
ICD	11-6, 11-4	requirement
implications*	65-1	requirement
indicators*	25-4	requirement
information	94-1, 59-1	data
initiator	89-4	hardware
instability*	33-2, 33-4, 33-1, 27-3, 33-4	requirement
instruction	32-5, 4-4, 64-1	requirement
instrument	55-2, 64-7, 94-6, 64-4, 94-5	hardware
insulator	8-3, 14-4, 42-5	hardware
interfaces	86-1, 76-2	hardware
interference*	86-2	requirement
item	72-3, 93-2, 78-3	hardware
joint	69-3, 4-5, 15-1	hardware
latch	44-5	hardware
launch vehicle	2-5, 1-2, 2-5, 33-3, 33-1, 44-5, 89-4, 91-6, 11-8, 83-2	segment
layer*	31-3	requirement
lead	39-5	hardware
lens	96-4	hardware
lines	39-4, 98-2, 90-8	hardware
loads	27-1, 30-2	hardware
logic path	25-3	software
loops	27-5	software
louvers	64-4, 64-6, 64-5	hardware
lubricant	9-1, 9-3, 9-4, 57-5, 9-2	hardware
lubrication*	21-2	requirement
magnet	60-5, 72-4	hardware
margins*	7-3, 66-1, 11-2, 37-4, 55-3, 66-1	requirement
Mariner	43-5	system
mass property*	2-1	requirement
material	14-2, 95-2, 90-1, 92-3, 14-1, 21-4, 22-2, 10-2, 21-4, 22-2, 45-1, 78-1, 92-3, 78-1, 92-4	hardware
mechanical*	68-1	requirement
mechanisms*	100-1, 100-2, 27-3, 100-1, 100-2	requirement
metal	31-1	hardware
mismatch*	52-2	requirement
missiles	86-3	subsystem
modes	41-2	software
modules	93-4	software
moisture*	21-2	requirement
monitors	39-4	hardware
motions*	20-2	requirement
motor	9-4, 50-5, 50-4, 56-4, 98-2	hardware
mount	47-4	hardware
needles	60-5	hardware
nitrogen	52-4	hardware

Original Concept	PR # - LL #	Coded Concept
nozzle	14-4	hardware
nozzle skirt	14-3	hardware
null	40-4	data
oil	21-1, 21-7, 21-2, 21-6, 21-7	hardware
orbit*	27-4	requirement
oscilloscopes	46-3	hardware
overlays	34-3	hardware
oxidation*	21-3	requirement
parameter*	79-1	requirement
particles*	17-3	requirement
parts	92-2, 66-4, 5-4, 21-3, 21-3, 92-2, 8-1	hardware
payload	49-4, 94-6	segment
performance*	23-4, 94-8	requirement
pin	9-3, 89-3	hardware
pipes	45-1, 30-3	hardware
plasma	98-4	hardware
plate	47-4	hardware
plug	32-5	hardware
Plug–and–Play*	62-2	requirement
polarity*	53-2, 53-5, 93-4	requirement
poles	60-5	data
port	44-6, 57-4, 46-4	hardware
power supplies	38-4	hardware
power*	37-2, 44-2	requirement
precision*	48-2, 48-4	requirement
probe	90-9, 23-6	hardware
problems	51-3, 80-1	data
procedure*	32-2, 76-1, 2-3, 17-4, 54-1, 29-4, 54-1, 63-1, 76-1, 88-3, 17-4, 29-4, 54-1	requirement
processor	50-5, 35-2, 36-4, 94-5	hardware
product	99-3, 99-4, 26-1	subsystem
project stores	5-3	software
propellant	27-2, 27-3, 27-4	hardware
proprietary data	2-6, 23-3	data
protection*	35-2, 30-2, 34-1, 35-1, 36-3	requirement
pulses	36-4	data
pyros	7-1	hardware
radar	48-4	hardware
radiation	14-6	requirement
radiators	16-4	hardware
rags	90-8	hardware
range gate	48-4	hardware
rate	14-2	requirement
reactions*	22-4	requirement
regulator	32-5, 27-4, 32-5, 27-4	hardware
relay	44-5, 44-6, 77-2, 71-1, 72-4	hardware
reports	69-2	requirement
requirement*	4-3, 12-1, 12-4, 12-5, 12-1, 12-1, 12-2, 12-4, 33-3, 76-3, 23-1, 5-2, 23-6, 60-2, 19-1, 4-1, 4-2, 12-2, 40-1, 56-2, 60-2, 67-4, 73-4, 85-2, 100-3, 73-3, 73-4, 85-2	requirement
resin	14-3	hardware
resistor	98-5, 56-5, 19-6	hardware
resonance	11-5, 11-6	requirement
ring	41-3, 41-4, 56-4	hardware
rocket	6-2, 6-3, 14-5, 87-3, 87-3, 6-3	subsystem
rod	96-4, 96-3	hardware
routine	36-4	software
rubber spacer	14-6	hardware
safety*	65-2	requirement
satellite	39-2,8-2,11-7,30-1,67-1,20-4,51-7,11-5,11-8,17-3,17-6,24-3,30-1,30-2,33-1,33-3,39-2,44-5,51-7,53-4,64-4,67-1,67-2,67-3,,73-1,88-5,88-6,20-4,88-5,17-3,90-9,33-4,65-	segment

Original Concept	PR # - LL #	Coded Concept
	3,88-6,10-2,27-4,57-5	
scenario*	55-4,36-1	requirement
screw	32-5,47-4	hardware
Scud	48-4	system
seal	15-4,57-5	hardware
semiconductor	31-1	hardware
sensor	50-5,44-5,56-4,94-7,67-3	hardware
separators	22-3	hardware
shield	81-3,88-5,88-6	hardware
shielding*	10-2	requirement
side	88-5, 63-3	data
signals	44-2,40-4,44-2	data
signs	89-3,65-6	data
silicone	51-6,51-7	hardware
simulator	29-1	subsystem
sleeves	15-2	hardware
socket	9-3	hardware
software	3-2,23-3,25-2,29-1,43-5,12-4,12-6,32-3,35-3,36-2,53-1,60-2,73-2,3-1,18-2,18-4,18-7,19-4,25-1,43-2,43-5,3-2,12-4,12-6,18-1,18-2,18-3,18-4,18-6,18-6,23-3,29-1,30-3,35-1,43-5, 43-5,46-5,60-2,68-1,73-1,79-3,79-5,85-3,94-1,94-7,32-3,79-3,85-3,30-3,46-5,48-1,94-1,94-7,12-2,36-2,53-1	software
solar array	23-2	hardware
solar panels	8-1,14-6,8-1,13-1,13-2,13-1,13-2,13-3,17-3,53-4,44-5,81-3,100-1	hardware
solvent	14-5	hardware
source code	3-4,43-1	software
sources	43-3,91-2,86-2	data
space environment	17-1	requirement
space system	1-1	system
space vehicle	2-5	segment
space*	20-5,64-7	requirement
spacecraft	10-1,13-4,29-3,1-2,11-7,13-4,13-5,49-5,49-5,71-4,79-1	segment
specification*	37-1,11-4,37-1,18-3,18-3,5-4,54-6,73-3	requirement
spectrum	40-2	data
speed	11-4	requirement
stability control	2-1-G	requirement
state–of–the–art	52-3	data
station	53-3	subsystem
storm*	17-5,17-2,17-6	requirement
stress	11-1,8-3	requirement
strips	72-4	hardware
structural loads	2-1	requirement
structure	10-2,89-4,28-2,28-4	subsystem
subsystem	97-5,12-2,66-3,97-5,95-3,71-4,97-2,17-4,71-4,17-4	subsystem
sunshield	42-5	hardware
surfaces	5-5,	requirement
switch	86-1	hardware
system	93-1,42-4,3-6,11-1,71-1,19-6,36-1,36-3,37-2,49-4,57-4,64-5,71-1,73-4,93-1,94-1,94-2,49-4,36-3,64-5,93-1,94-3,18-2,18-5,37-1,70-1,71-2,85-3,52-1,57-2,7-1,11-1,41-2,52-1,57-2,79-3,91-2,91-3	system
tables*	77-1	requirement
tank	27-2,30-3,27-4,30-3,27-5	hardware
tape	88-5,51-6,65-3,51-5,65-3,51-4,88-5,65-3	hardware
tape drive	24-6	hardware
TBD*	76-5, 76-4	requirement
technology	92-1,87-1,37-4	requirement
telemetry	67-1	data
telescopes	17-3	hardware
temperature	1-3	data

Original Concept	PR # - LL #	Coded Concept
thermal*	13-1,24-2	requirement
thermostat	30-3	hardware
thermostats	71-1,71-3	hardware
thread	15-4,32-5	software
thruster	57-5,17-6,30-3	hardware
timer	36-4	hardware
titanium	31-4	hardware
tolerances*	52-2	requirement
tools	16-2,90-10,96-1	hardware
torque*	53-2,53-5,60-4,15-3,15-2	requirement
torquers	53-6,53-5	hardware
tracker	80-5	hardware
transistors	56-5	hardware
tube	72-4,45-5,45-5	hardware
tunnel	81-3	hardware
unit	82-2,84-4,86-1,18-5,36-1,66-3,82-2,84-4,86-1,91-5,18-5,90-10,89-1,64-1,73-3,84-1,89-1,73-3,57-5,84-2	hardware
vacuum*	49-2,1-3	requirement
valve	57-4,57-4,83-2,54-6,14-5,27-4,39-4,83-2	hardware
vehicle	67-1,73-1,66-3,66-1,66-3,67-1,73-1,89-4,91-6,91-6	segment
vent	54-4	hardware
vessel	54-3,28-1	subsystem
vibration*	11-8,11-4,11-6,11-8,24-3,24-4,24-3,24-4,11-7	requirement
voltage*	31-3, 24-6	requirement
volume*	37-2	requirement
walls	47-5	hardware
warm days*	59-5	requirement
washers	96-5,96-4	hardware
weaknesses*	91-3	requirement
weight*	37-2	requirement
wheels	21-6	hardware
wires	62-4,78-1,91-6,39-5,91-2,39-5	hardware

Appendix C: Traceability Matrix-Model

Table 14. Traceability Matrix-Model

		Phase 0: Concept Studies	Alternative System Review (ASR)	Phase A: Concept Development	Integrated Baseline Review (IBR)	System Requirements Review (SRR)	System Design Review (SDR)	Phase B: Preliminary Design	Preliminary Design Review (PDR)	Phase C: Complete Design	Critical Design Review (CDR)	Phase D1: Fabrication & Integration	Production Readiness Review (PRR)	Test Readiness Review (TRR)	System Verification Review (SVR)	Functional Configuration Audit (FCA)	Physical Configuration Audit (PCA)	Phase D2: Fielding & Checkout	Mission Readiness Review (MRR)	Flight Readiness Review (FRR)	Launch Readiness Review (LRR)	Phase D3: Operations & Disposal	Post Flight Review (PFR)
A: People-People																							
10	Acquirer-Developer				X	X	X		X		X		X	X	X	X	X		X	X	X		X
4	Developer (Launcher)-Developer (Satellite)					X	X		X		X		X	X	X	X	X		X	X	X		
3	Developer-Supplier						X		X		X		X	X	X	X	X		X	X	X		
2	Developer (Payload)-Developer (Bus)					X	X		X		X		X	X	X	X	X		X	X	X		
1	Acquirer-Stakeholder (Independent)		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
1	Developer (Sub)-Developer (Prime)					X	X		X		X		X	X	X	X	X		X	X	X		
1	Developer (System)-Developer (Software)					X	X		X		X		X	X	X	X	X		X	X	X		
1	Developer-Developer					X	X		X		X		X	X	X	X	X		X	X	X		
1	Developer-Operator				X	X	X		X		X		X	X	X	X	X		X	X	X		
1	Stakeholder (AFSPC)-Stakeholder (Public)		X		X						X		X	X	X	X	X		X	X	X		X
25	**TOTAL**		2		4	8	9		9		10		10	10	10	10	10		10	10	10		3
B: Process-Process																							
26	Evaluate-Deploy										X		X	X	X	X	X		X	X	X		X
17	Design-Evaluate						X		X		X		X	X	X	X	X		X	X	X		X
13	Define-Evaluate				X	X	X		X		X		X	X	X	X	X		X	X	X		X
13	Design-Deploy						X		X		X		X	X	X	X	X		X	X	X		X
13	Evaluate-Evaluate		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
13	Implement-Evaluate										X		X	X	X	X	X		X	X	X		X
12	Deploy-Deploy												X	X	X	X	X		X	X	X		X

		Phase 0: Concept Studies	Alternative System Review (ASR)	Phase A: Concept Development	Integrated Baseline Review (IBR)	System Requirements Review (SRR)	System Design Review (SDR)	Phase B: Preliminary Design	Preliminary Design Review (PDR)	Phase C: Complete Design	Critical Design Review (CDR)	Phase D1: Fabrication & Integration	Production Readiness Review (PRR)	Test Readiness Review (TRR)	System Verification Review (SVR)	Functional Configuration Audit (FCA)	Physical Configuration Audit (PCA)	Phase D2: Fielding & Checkout	Mission Readiness Review (MRR)	Flight Readiness Review (FRR)	Launch Readiness Review (LRR)	Phase D3: Operations & Disposal	Post Flight Review (PFR)
9	Deploy-Manage												X	X	X	X	X		X	X	X		X
8	Evaluate-Manage		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
8	Implement-Deploy										X		X	X	X	X	X		X	X	X		X
6	Implement-Manage										X		X	X	X	X	X		X	X	X		X
5	Define-Implement				X	X	X		X		X		X	X	X	X	X		X	X	X		X
5	Design-Evaluate-Deploy								X		X		X	X	X	X	X		X	X	X		X
5	Design-Implement								X		X		X	X	X	X	X		X	X	X		X
5	Design-Manage		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
5	Implement-Evaluate-Deploy										X		X	X	X	X	X		X	X	X		X
4	Define-Implement-Evaluate		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
4	Implement-Implement								X		X		X	X	X	X	X		X	X	X		X
3	Define-Define		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
3	Define-Design				X	X	X		X		X		X	X	X	X	X		X	X	X		X
3	Define-Design-Implement		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
2	Define-Deploy				X	X	X		X		X		X	X	X	X	X		X	X	X		X
2	Define-Evaluate-Deploy		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
2	Design-Implement-Deploy										X		X	X	X	X	X		X	X	X		X
2	Design-Implement-Evaluate-Deploy										X		X	X	X	X	X		X	X	X		X
2	Evaluate-Deploy-Manage										X		X	X	X	X	X		X	X	X		X
2	Implement-Evaluate-Manage										X		X	X	X	X	X		X	X	X		X
2	Manage-Manage		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
1	Define-Design-Evaluate-Deploy										X		X	X	X	X	X		X	X	X		X
1	Define-Design-Implement-Evaluate										X		X	X	X	X	X		X	X	X		X
1	Define-Implement-Deploy		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
1	Define-Implement-Evaluate-Deploy										X		X	X	X	X	X		X	X	X		X

		Phase 0: Concept Studies	Alternative System Review (ASR)	Phase A: Concept Development	Integrated Baseline Review (IBR)	System Requirements Review (SRR)	System Design Review (SDR)	Phase B: Preliminary Design	Preliminary Design Review (PDR)	Phase C: Complete Design	Critical Design Review (CDR)	Phase D1: Fabrication & Integration	Production Readiness Review (PRR)	Test Readiness Review (TRR)	System Verification Review (SVR)	Functional Configuration Audit (FCA)	Physical Configuration Audit (PCA)	Phase D2: Fielding & Checkout	Mission Readiness Review (MRR)	Flight Readiness Review (FRR)	Launch Readiness Review (LRR)	Phase D3: Operations & Disposal	Post Flight Review (PFR)
1	Define-Manage		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
1	Design-Design		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
1	Design-Implement-Deploy-Manage										X		X	X	X	X	X		X	X	X		X
1	Design-Implement-Evaluate										X		X	X	X	X	X		X	X	X		X
1	Implement-Deploy-Manage										X		X	X	X	X	X		X	X	X		X
1	Implement-Evaluate-Deploy-Manage										X		X	X	X	X	X		X	X	X		X
204	**TOTAL**		11		15	15	17		20		36		38	38	38	38	38		38	38	38		38
C: Product-Product																							
62	Hardware-Hardware								X		X		X	X	X	X	X		X	X	X		X
38	Hardware-Requirement					X	X		X		X		X	X	X	X	X		X	X	X		X
17	Segment-Hardware						X		X		X		X	X	X	X	X		X	X	X		X
13	System-Hardware						X		X		X		X	X	X	X	X		X	X	X		X
12	Hardware-Software								X		X		X	X	X	X	X		X	X	X		X
9	Hardware-Data								X		X		X	X	X	X	X		X	X	X		X
7	Segment-Requirement		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
5	Hardware-Software-Requirement					X	X		X		X		X	X	X	X	X		X	X	X		X
5	Requirement-Requirement		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
5	Software-Requirement					X	X		X		X		X	X	X	X	X		X	X	X		X
5	System-Requirement		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
4	Segment-Segment		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
4	Subsystem-Hardware								X		X		X	X	X	X	X		X	X	X		X
4	Subsystem-Requirement					X	X		X		X		X	X	X	X	X		X	X	X		X
4	System-Software						X		X		X		X	X	X	X	X		X	X	X		X
3	Data-Requirement		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
3	Segment-Hardware-Data						X		X		X		X	X	X	X	X		X	X	X		X

		Phase 0: Concept Studies	Alternative System Review (ASR)	Phase A: Concept Development	Integrated Baseline Review (IBR)	System Requirements Review (SRR)	System Design Review (SDR)	Phase B: Preliminary Design	Preliminary Design Review (PDR)	Phase C: Complete Design	Critical Design Review (CDR)	Phase D1: Fabrication & Integration	Production Readiness Review (PRR)	Test Readiness Review (TRR)	System Verification Review (SVR)	Functional Configuration Audit (FCA)	Physical Configuration Audit (PCA)	Phase D2: Fielding & Checkout	Mission Readiness Review (MRR)	Flight Readiness Review (FRR)	Launch Readiness Review (LRR)	Phase D3: Operations & Disposal	Post Flight Review (PFR)
3	Software-Software								X		X		X	X	X	X	X		X	X	X		X
3	System-Hardware-Requirement					X	X		X		X		X	X	X	X	X		X	X	X		X
2	Hardware-Software-Data								X		X		X	X	X	X	X		X	X	X		X
2	Segment-Hardware-Requirement					X	X		X		X		X	X	X	X	X		X	X	X		X
2	Segment-Software						X		X		X		X	X	X	X	X		X	X	X		X
2	Segment-Subsystem-Hardware						X		X		X		X	X	X	X	X		X	X	X		X
1	Data-Data				X	X	X		X		X		X	X	X	X	X		X	X	X		X
1	Hardware-Data-Requirement					X	X		X		X		X	X	X	X	X		X	X	X		X
1	Segment-Subsystem					X	X		X		X		X	X	X	X	X		X	X	X		X
1	Subsystem-Hardware-Requirement						X		X		X		X	X	X	X	X		X	X	X		X
1	Subsystem-Hardware-Software								X		X		X	X	X	X	X		X	X	X		X
1	Subsystem-Software								X		X		X	X	X	X	X		X	X	X		X
1	Subsystem-Software-Requirement					X	X		X		X		X	X	X	X	X		X	X	X		X
1	System-Hardware-Data						X		X		X		X	X	X	X	X		X	X	X		X
1	System-Hardware-Software						X		X		X		X	X	X	X	X		X	X	X		X
1	System-Segment		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
1	System-Segment-Hardware						X		X		X		X	X	X	X	X		X	X	X		X
1	System-Segment-Requirement		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
1	System-Software-Data						X		X		X		X	X	X	X	X		X	X	X		X
1	System-Subsystem-Requirement						X		X		X		X	X	X	X	X		X	X	X		X
228	**TOTAL**		7		8	17	29		37		37		37	37	37	37	37		37	37	37		37
D: People-Process																							
3	Developer-Evaluate								X		X		X	X	X	X	X		X	X	X		
2	Developer-Define					X	X		X		X		X	X	X	X	X		X	X	X		
2	Developer-Design						X		X		X		X	X	X	X	X		X	X	X		

		Phase 0: Concept Studies	Alternative System Review (ASR)	Phase A: Concept Development	Integrated Baseline Review (IBR)	System Requirements Review (SRR)	System Design Review (SDR)	Phase B: Preliminary Design	Preliminary Design Review (PDR)	Phase C: Complete Design	Critical Design Review (CDR)	Phase D1: Fabrication & Integration	Production Readiness Review (PRR)	Test Readiness Review (TRR)	System Verification Review (SVR)	Functional Configuration Audit (FCA)	Physical Configuration Audit (PCA)	Phase D2: Fielding & Checkout	Mission Readiness Review (MRR)	Flight Readiness Review (FRR)	Launch Readiness Review (LRR)	Phase D3: Operations & Disposal	Post Flight Review (PFR)
2	Developer-Implement								X		X		X	X	X	X	X		X	X	X		X
2	Developer-Manage						X		X		X		X	X	X	X	X		X	X	X		
1	Acquirer-Define		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
1	Acquirer-Manage		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
1	Operator-Define		X		X	X	X																X
1	Stakeholder (Independent)-Define		X		X	X	X																X
15	**TOTAL**		4		4	5	7		7		7		7	7	7	7	7		7	7	7		5
E: Process-Product																							
40	Deploy-Hardware												X	X	X	X	X		X	X	X		X
38	Evaluate-Hardware										X		X	X	X	X	X		X	X	X		X
24	Design-Hardware						X		X		X		X	X	X	X	X		X	X	X		X
20	Evaluate-Requirement				X	X	X		X		X		X	X	X	X	X		X	X	X		X
10	Design-Requirement					X	X		X		X		X	X	X	X	X		X	X	X		X
10	Implement-Hardware										X		X	X	X	X	X		X	X	X		X
10	Manage-Requirement		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
9	Define-Requirement		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
9	Deploy-Requirement				X	X	X		X		X		X	X	X	X	X		X	X	X		X
6	Deploy-Segment												X	X	X	X	X		X	X	X		X
6	Implement-Requirement				X	X	X		X		X		X	X	X	X	X		X	X	X		X
6	Manage-Hardware								X		X		X	X	X	X	X		X	X	X		X
5	Evaluate-Software								X		X		X	X	X	X	X		X	X	X		X
5	Evaluate-System		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
4	Deploy-Data												X	X	X	X	X		X	X	X		X
4	Design-Software						X		X		X		X	X	X	X	X		X	X	X		X
4	Evaluate-Data										X		X	X	X	X	X		X	X	X		X

		Phase 0: Concept Studies	Alternative System Review (ASR)	Phase A: Concept Development	Integrated Baseline Review (IBR)	System Requirements Review (SRR)	System Design Review (SDR)	Phase B: Preliminary Design	Preliminary Design Review (PDR)	Phase C: Complete Design	Critical Design Review (CDR)	Phase D1: Fabrication & Integration	Production Readiness Review (PRR)	Test Readiness Review (TRR)	System Verification Review (SVR)	Functional Configuration Audit (FCA)	Physical Configuration Audit (PCA)	Phase D2: Fielding & Checkout	Mission Readiness Review (MRR)	Flight Readiness Review (FRR)	Launch Readiness Review (LRR)	Phase D3: Operations & Disposal	Post Flight Review (PFR)
4	Evaluate-Segment		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
4	Implement-Software										X		X	X	X	X	X		X	X	X		X
3	Define-Hardware								X		X		X	X	X	X	X		X	X	X		X
3	Deploy-Software												X	X	X	X	X		X	X	X		X
3	Design-Evaluate-Hardware												X	X	X	X	X		X	X	X		X
3	Design-Segment				X	X	X		X		X		X	X	X	X	X		X	X	X		X
3	Design-System				X	X	X		X		X		X	X	X	X	X		X	X	X		X
3	Evaluate-Deploy-Hardware												X	X	X	X	X		X	X	X		X
3	Evaluate-Hardware-Software										X		X	X	X	X	X		X	X	X		X
2	Define-Data				X	X	X		X		X		X	X	X	X	X		X	X	X		X
2	Deploy-Subsystem												X	X	X	X	X		X	X	X		X
2	Evaluate-Deploy-Requirement										X		X	X	X	X	X		X	X	X		X
2	Evaluate-Hardware-Data										X		X	X	X	X	X		X	X	X		X
2	Evaluate-Software-Requirement								X		X		X	X	X	X	X		X	X	X		X
2	Evaluate-Subsystem					X	X		X		X		X	X	X	X	X		X	X	X		X
2	Manage-Data				X	X	X		X		X		X	X	X	X	X		X	X	X		X
1	Define-Data-Requirement				X	X	X		X		X		X	X	X	X	X		X	X	X		X
1	Define-Design-Implement-Evaluate-Rqmt										X		X	X	X	X	X		X	X	X		X
1	Define-Design-Requirement					X	X		X		X		X	X	X	X	X		X	X	X		X
1	Define-Design-Software						X		X		X		X	X	X	X	X		X	X	X		X
1	Define-Evaluate-Hardware										X		X	X	X	X	X		X	X	X		X
1	Define-Hardware-Requirement						X		X		X		X	X	X	X	X		X	X	X		X
1	Define-Implement-Data										X		X	X	X	X	X		X	X	X		X
1	Define-Implement-Evaluate-Hardware										X		X	X	X	X	X		X	X	X		X
1	Define-Implement-Software										X		X	X	X	X	X		X	X	X		X

		Phase 0: Concept Studies	Alternative System Review (ASR)	Phase A: Concept Development	Integrated Baseline Review (IBR)	System Requirements Review (SRR)	System Design Review (SDR)	Phase B: Preliminary Design	Preliminary Design Review (PDR)	Phase C: Complete Design	Critical Design Review (CDR)	Phase D1: Fabrication & Integration	Production Readiness Review (PRR)	Test Readiness Review (TRR)	System Verification Review (SVR)	Functional Configuration Audit (FCA)	Physical Configuration Audit (PCA)	Phase D2: Fielding & Checkout	Mission Readiness Review (MRR)	Flight Readiness Review (FRR)	Launch Readiness Review (LRR)	Phase D3: Operations & Disposal	Post Flight Review (PFR)
1	Define-Software								X		X		X	X	X	X	X		X	X	X		X
1	Deploy-Evaluate-Hardware-Requirement										X		X	X	X	X	X		X	X	X		X
1	Deploy-System-Segment-Requirement				X	X	X		X		X		X	X	X	X	X		X	X	X		X
1	Design-Data						X		X		X		X	X	X	X	X		X	X	X		X
1	Design-Deploy-Segment						X		X		X		X	X	X	X	X		X	X	X		X
1	Design-Deploy-Software												X	X	X	X	X		X	X	X		X
1	Design-Evaluate-System-Hardware-Rqmt					X	X		X		X		X	X	X	X	X		X	X	X		X
1	Design-Hardware-Data						X		X		X		X	X	X	X	X		X	X	X		X
1	Design-Implement-Deploy-Mng-Hardware												X	X	X	X	X		X	X	X		X
1	Design-Implement-Hardware										X		X	X	X	X	X		X	X	X		X
1	Design-Segment-Requirement				X	X	X		X		X		X	X	X	X	X		X	X	X		X
1	Design-Subsystem						X		X		X		X	X	X	X	X		X	X	X		X
1	Evaluate-Deploy-Data-Requirement												X	X	X	X	X		X	X	X		X
1	Evaluate-Deploy-Hardware-Requirement												X	X	X	X	X		X	X	X		X
1	Evaluate-Hardware-Requirement					X	X		X		X		X	X	X	X	X		X	X	X		X
1	Evaluate-Manage-Requirement				X	X	X		X		X		X	X	X	X	X		X	X	X		X
1	Implement-Evaluate-Hardware-Rqmt										X		X	X	X	X	X		X	X	X		X
1	Implement-Evaluate-Software										X		X	X	X	X	X		X	X	X		X
1	Implement-Evaluate-Software-Requirement										X		X	X	X	X	X		X	X	X		X
1	Implement-Hardware-Requirement										X		X	X	X	X	X		X	X	X		X
1	Implement-Segment						X		X		X		X	X	X	X	X		X	X	X		X
1	Implement-Software-Requirement										X		X	X	X	X	X		X	X	X		X
1	Implement-Subsystem						X		X		X		X	X	X	X	X		X	X	X		X
1	Implement-System					X	X		X		X		X	X	X	X	X		X	X	X		X
1	Manage-Data-Requirement				X	X	X		X		X		X	X	X	X	X		X	X	X		X

		Phase 0: Concept Studies	Alternative System Review (ASR)	Phase A: Concept Development	Integrated Baseline Review (IBR)	System Requirements Review (SRR)	System Design Review (SDR)	Phase B: Preliminary Design	Preliminary Design Review (PDR)	Phase C: Complete Design	Critical Design Review (CDR)	Phase D1: Fabrication & Integration	Production Readiness Review (PRR)	Test Readiness Review (TRR)	System Verification Review (SVR)	Functional Configuration Audit (FCA)	Physical Configuration Audit (PCA)	Phase D2: Fielding & Checkout	Mission Readiness Review (MRR)	Flight Readiness Review (FRR)	Launch Readiness Review (LRR)	Phase D3: Operations & Disposal	Post Flight Review (PFR)
1	Manage-Hardware-Software								X		X		X	X	X	X	X		X	X	X		X
1	Manage-Software								X		X		X	X	X	X	X		X	X	X		X
1	Manage-Subsystem						X		X		X		X	X	X	X	X		X	X	X		X
1	Manage-System		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
291	TOTAL		5		17	23	34		41		60		71	71	71	71	71		71	71	71		71
F: People-Product																							
2	Supplier-Requirements								X		X		X	X	X	X	X		X	X	X		
1	Acquirer-Data		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
1	Developer-Data						X		X		X		X	X	X	X	X		X	X	X		
1	Developer-Requirements						X		X		X		X	X	X	X	X		X	X	X		
1	Operator-System		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
1	Stakeholder-Hardware		X		X	X					X		X	X	X	X	X		X	X	X		X
7	TOTAL		3		3	3	4		5		6		6	6	6	6	6		6	6	6		3
G: People-Process-Product																							
7	Developer-Evaluate-Hardware								X		X		X	X	X	X	X		X	X	X		X
7	Developer-Implement-Hardware								X		X		X	X	X	X	X		X	X	X		X
4	Developer-Design-Hardware								X		X		X	X	X	X	X		X	X	X		X
4	Developer-Manage-Requirement					X	X		X		X		X	X	X	X	X		X	X	X		X
3	Developer-Define-Requirement					X	X		X		X		X	X	X	X	X		X	X	X		X
3	Developer-Deploy-Hardware										X		X	X	X	X	X		X	X	X		X
3	Developer-Design-Requirement					X	X		X		X		X	X	X	X	X		X	X	X		X
3	Developer-Evaluate-Requirement					X	X		X		X		X	X	X	X	X		X	X	X		X
3	Operator-Deploy-Hardware												X	X	X	X	X		X	X	X		X
2	Developer-Deploy-Requirement						X		X		X		X	X	X	X	X		X	X	X		X

		Phase 0: Concept Studies	Alternative System Review (ASR)	Phase A: Concept Development	Integrated Baseline Review (IBR)	System Requirements Review (SRR)	System Design Review (SDR)	Phase B: Preliminary Design	Preliminary Design Review (PDR)	Phase C: Complete Design	Critical Design Review (CDR)	Phase D1: Fabrication & Integration	Production Readiness Review (PRR)	Test Readiness Review (TRR)	System Verification Review (SVR)	Functional Configuration Audit (FCA)	Physical Configuration Audit (PCA)	Phase D2: Fielding & Checkout	Mission Readiness Review (MRR)	Flight Readiness Review (FRR)	Launch Readiness Review (LRR)	Phase D3: Operations & Disposal	Post Flight Review (PFR)
2	Developer-Design-Software								X		X		X	X	X	X	X		X	X	X		X
2	Developer-Evaluate-Subsystem								X		X		X	X	X	X	X		X	X	X		X
2	Developer-Manage-Hardware								X		X		X	X	X	X	X		X	X	X		X
2	Operator-Deploy-Requirement		X		X						X		X	X	X	X	X		X	X	X		X
2	Operator-Evaluate-Requirement		X		X						X		X	X	X	X	X		X	X	X		X
2	Supplier-Implement-Hardware								X		X		X	X	X	X	X		X	X	X		X
1	Acquirer-Evaluate-Requirement		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
1	Acquirer-Evaluate-Subsystem						X		X		X		X	X	X	X	X		X	X	X		X
1	Acquirer-Implement-Software								X		X		X	X	X	X	X		X	X	X		X
1	Acquirer-Manage-Requirement		X		X	X	X		X		X		X	X	X	X	X		X	X	X		X
1	Developer-Define-Data					X	X		X		X		X	X	X	X	X		X	X	X		X
1	Developer-Define-Evaluate-Requirement					X	X		X		X		X	X	X	X	X		X	X	X		X
1	Developer-Define-Manage-Requirement					X	X		X		X		X	X	X	X	X		X	X	X		X
1	Developer-Deploy-Segment						X		X		X		X	X	X	X	X		X	X	X		X
1	Developer-Deploy-Software										X		X	X	X	X	X		X	X	X		X
1	Developer-Design-Data						X		X		X		X	X	X	X	X		X	X	X		X
1	Developer-Design-Hardware-Software								X		X		X	X	X	X	X		X	X	X		X
1	Developer-Evaluate-Data								X		X		X	X	X	X	X		X	X	X		X
1	Developer-Evaluate-Hardware-Rqmt						X		X		X		X	X	X	X	X		X	X	X		X
1	Developer-Evaluate-Hardware-Software								X		X		X	X	X	X	X		X	X	X		X
1	Developer-Evaluate-Software								X		X		X	X	X	X	X		X	X	X		X
1	Developer-Evaluate-System					X	X		X		X		X	X	X	X	X		X	X	X		X
1	Developer-Evaluate-System-Hardware						X		X		X		X	X	X	X	X		X	X	X		X
1	Developer-Implement-Evaluate-Mng-SW								X		X		X	X	X	X	X		X	X	X		X
1	Developer-Implement-Requirement					X	X		X		X		X	X	X	X	X		X	X	X		X

		Phase 0: Concept Studies	Alternative System Review (ASR)	Phase A: Concept Development	Integrated Baseline Review (IBR)	System Requirements Review (SRR)	System Design Review (SDR)	Phase B: Preliminary Design	Preliminary Design Review (PDR)	Phase C: Complete Design	Critical Design Review (CDR)	Phase D1: Fabrication & Integration	Production Readiness Review (PRR)	Test Readiness Review (TRR)	System Verification Review (SVR)	Functional Configuration Audit (FCA)	Physical Configuration Audit (PCA)	Phase D2: Fielding & Checkout	Mission Readiness Review (MRR)	Flight Readiness Review (FRR)	Launch Readiness Review (LRR)	Phase D3: Operations & Disposal	Post Flight Review (PFR)
1	Developer-Implement-Software								X		X		X	X	X	X	X		X	X	X		X
1	Developer-Manage-Data					X	X		X		X		X	X	X	X	X		X	X	X		X
1	Developer-Operator-Deploy-Requirement										X		X	X	X	X	X		X	X	X		X
1	Operator-Define-Requirement		X		X						X		X	X	X	X	X		X	X	X		X
1	Operator-Evaluate-Data										X		X	X	X	X	X		X	X	X		X
1	Operator-Manage-Hardware												X	X	X	X	X		X	X	X		X
1	Operator-Manage-Requirement		X		X	X							X	X	X	X	X		X	X	X		X
1	Stakeholder-Define-Segment		X		X						X		X	X	X	X	X		X	X	X		X
1	Stakeholder-Deploy-Hardware												X	X	X	X	X		X	X	X		X
1	Stakeholder-Evaluate-Requirement		X		X						X		X	X	X	X	X		X	X	X		X
1	Stakeholder-Manage-Requirement		X		X	X							X	X	X	X	X		X	X	X		X
1	Supplier-Define-Hardware								X		X		X	X	X	X	X		X	X	X		X
1	Supplier-Evaluate-Hardware								X		X		X	X	X	X	X		X	X	X		X
1	Supplier-Evaluate-Requirement								X		X		X	X	X	X	X		X	X	X		X
1	User-Deploy-Data												X	X	X	X	X		X	X	X		X
85	**TOTAL**		9		9	14	18		35		44		50	50	50	50	50		50	50	50		50

Table 15. Traceability Matrix for GPS Case Study

		Phase 0: Concept Studies	Alternative System Review (ASR)	Phase A: Concept Development	Integrated Baseline Review (IBR)	System Requirements Review (SRR)	System Design Review (SDR)	Phase B: Preliminary Design	Preliminary Design Review (PDR)	Phase C: Complete Design	Critical Design Review (CDR)	Phase D1: Fabrication & Integration	Production Readiness Review (PRR)	Test Readiness Review (TRR)	System Verification Review (SVR)	Functional Configuration Audit (FCA)	Physical Configuration Audit (PCA)	Phase D2: Fielding & Checkout	Mission Readiness Review (MRR)	Flight Readiness Review (FRR)	Launch Readiness Review (LRR)	Phase D3: Operations & Disposal	Post Flight Review (PFR)
A: People-People																							
10	Acquirer-Developer				X		X		X		X		X			X							
4	Developer (Launcher)-Developer (Satellite)												X			X							
3	Developer-Supplier				X																		
2	Developer (Payload)-Developer (Bus)												X										
1	Acquirer-Stakeholder (Independent)				X						X		X			X							
1	Developer (Sub)-Developer (Prime)				X	X							X			X							
1	Developer (System)-Developer (Software)				X	X	X									X							
1	Developer-Developer						X		X		X												
1	Developer-Operator						X				X												
1	Stakeholder (AFSPC)-Stakeholder (Public)																						
25	**TOTAL**		0		5	2	4		2		4		5	0	0	5	0		0	0	0		0
B: Process-Process																							
26	Evaluate-Deploy																						
17	Design-Evaluate					X											X						
13	Define-Evaluate					X	X		X		X		X			X							
13	Design-Deploy						X		X							X							
13	Evaluate-Evaluate						X		X														
13	Implement-Evaluate																						
12	Deploy-Deploy																						
9	Deploy-Manage																						

		Phase 0: Concept Studies	Alternative System Review (ASR)	Phase A: Concept Development	Integrated Baseline Review (IBR)	System Requirements Review (SRR)	System Design Review (SDR)	Phase B: Preliminary Design	Preliminary Design Review (PDR)	Phase C: Complete Design	Critical Design Review (CDR)	Phase D1: Fabrication & Integration	Production Readiness Review (PRR)	Test Readiness Review (TRR)	System Verification Review (SVR)	Functional Configuration Audit (FCA)	Physical Configuration Audit (PCA)	Phase D2: Fielding & Checkout	Mission Readiness Review (MRR)	Flight Readiness Review (FRR)	Launch Readiness Review (LRR)	Phase D3: Operations & Disposal	Post Flight Review (PFR)
8	Evaluate-Manage												X										
8	Implement-Deploy																						
6	Implement-Manage				X		X		X														
5	Define-Implement				X	X					X					X							
5	Design-Evaluate-Deploy				X	X					X												
5	Design-Implement				X		X		X								X						
5	Design-Manage																X						
5	Implement-Evaluate-Deploy																						
4	Define-Implement-Evaluate				X				X														
4	Implement-Implement				X																		
3	Define-Define				X																		
3	Define-Design				X	X	X		X		X		X										
3	Define-Design-Implement										X												
2	Define-Deploy																						
2	Define-Evaluate-Deploy								X														
2	Design-Implement-Deploy					X					X												
2	Design-Implement-Evaluate-Deploy																						
2	Evaluate-Deploy-Manage																						
2	Implement-Evaluate-Manage																						
2	Manage-Manage																						
1	Define-Design-Evaluate-Deploy																						
1	Define-Design-Implement-Evaluate												X										
1	Define-Implement-Deploy				X				X														
1	Define-Implement-Evaluate-Deploy												X										
1	Define-Manage				X								X			X							

		Phase 0: Concept Studies	Alternative System Review (ASR)	Phase A: Concept Development	Integrated Baseline Review (IBR)	System Requirements Review (SRR)	System Design Review (SDR)	Phase B: Preliminary Design	Preliminary Design Review (PDR)	Phase C: Complete Design	Critical Design Review (CDR)	Phase D1: Fabrication & Integration	Production Readiness Review (PRR)	Test Readiness Review (TRR)	System Verification Review (SVR)	Functional Configuration Audit (FCA)	Physical Configuration Audit (PCA)	Phase D2: Fielding & Checkout	Mission Readiness Review (MRR)	Flight Readiness Review (FRR)	Launch Readiness Review (LRR)	Phase D3: Operations & Disposal	Post Flight Review (PFR)
1	Design-Design				X								X										
1	Design-Implement-Deploy-Manage																						
1	Design-Implement-Evaluate					X					X												
1	Implement-Deploy-Manage																						
1	Implement-Evaluate-Deploy-Manage																						
204	**TOTAL**		0		11	7	6		9		7		7	0	0	4	3		0	0	0		0
C: Product-Product																							
62	Hardware-Hardware				X																		
38	Hardware-Requirement				X								X				X						
17	Segment-Hardware										X		X										
13	System-Hardware						X				X		X										
12	Hardware-Software				X												X						
9	Hardware-Data										X												
7	Segment-Requirement																						
5	Hardware-Software-Requirement				X						X						X						
5	Requirement-Requirement												X										
5	Software-Requirement				X	X	X						X										
5	System-Requirement																						
4	Segment-Segment																						
4	Subsystem-Hardware										X												
4	Subsystem-Requirement				X																		
4	System-Software																						
3	Data-Requirement																						
3	Segment-Hardware-Data																						
3	Software-Software						X																

		Phase 0: Concept Studies	Alternative System Review (ASR)	Phase A: Concept Development	Integrated Baseline Review (IBR)	System Requirements Review (SRR)	System Design Review (SDR)	Phase B: Preliminary Design	Preliminary Design Review (PDR)	Phase C: Complete Design	Critical Design Review (CDR)	Phase D1: Fabrication & Integration	Production Readiness Review (PRR)	Test Readiness Review (TRR)	System Verification Review (SVR)	Functional Configuration Audit (FCA)	Physical Configuration Audit (PCA)	Phase D2: Fielding & Checkout	Mission Readiness Review (MRR)	Flight Readiness Review (FRR)	Launch Readiness Review (LRR)	Phase D3: Operations & Disposal	Post Flight Review (PFR)
3	System-Hardware-Requirement						X										X						
2	Hardware-Software-Data				X												X						
2	Segment-Hardware-Requirement												X										
2	Segment-Software				X						X		X										
2	Segment-Subsystem-Hardware										X		X										
1	Data-Data																						
1	Hardware-Data-Requirement										X												
1	Segment-Subsystem																						
1	Subsystem-Hardware-Requirement																						
1	Subsystem-Hardware-Software										X												
1	Subsystem-Software												X										
1	Subsystem-Software-Requirement						X																
1	System-Hardware-Data						X																
1	System-Hardware-Software						X										X						
1	System-Segment										X												
1	System-Segment-Hardware																						
1	System-Segment-Requirement																						
1	System-Software-Data																						
1	System-Subsystem-Requirement										X												
228	**TOTAL**		0		8	1	7		0		11		9	0	0	0	6		0	0	0		0
D: People-Process																							
3	Developer-Evaluate				X		X						X			X							
2	Developer-Define						X						X										
2	Developer-Design						X						X										
2	Developer-Implement				X								X			X							

		Phase 0: Concept Studies	Alternative System Review (ASR)	Phase A: Concept Development	Integrated Baseline Review (IBR)	System Requirements Review (SRR)	System Design Review (SDR)	Phase B: Preliminary Design	Preliminary Design Review (PDR)	Phase C: Complete Design	Critical Design Review (CDR)	Phase D1: Fabrication & Integration	Production Readiness Review (PRR)	Test Readiness Review (TRR)	System Verification Review (SVR)	Functional Configuration Audit (FCA)	Physical Configuration Audit (PCA)	Phase D2: Fielding & Checkout	Mission Readiness Review (MRR)	Flight Readiness Review (FRR)	Launch Readiness Review (LRR)	Phase D3: Operations & Disposal	Post Flight Review (PFR)
2	Developer-Manage				X						X		X										
1	Acquirer-Define																						
1	Acquirer-Manage										X												
1	Operator-Define																						
1	Stakeholder (Independent)-Define																						
15	**TOTAL**		0		3	0	3		0		2		5	0	0	2	0		0	0	0		0
E: Process-Product																							
40	Deploy-Hardware																						
38	Evaluate-Hardware																						
24	Design-Hardware																						
20	Evaluate-Requirement																						
10	Design-Requirement																						
10	Implement-Hardware																						
10	Manage-Requirement																						
9	Define-Requirement																						
9	Deploy-Requirement																						
6	Deploy-Segment																						
6	Implement-Requirement																						
6	Manage-Hardware																						
5	Evaluate-Software																						
5	Evaluate-System																						
4	Deploy-Data																						
4	Design-Software																						
4	Evaluate-Data																						
4	Evaluate-Segment																						

		Phase 0: Concept Studies	Alternative System Review (ASR)	Phase A: Concept Development	Integrated Baseline Review (IBR)	System Requirements Review (SRR)	System Design Review (SDR)	Phase B: Preliminary Design	Preliminary Design Review (PDR)	Phase C: Complete Design	Critical Design Review (CDR)	Phase D1: Fabrication & Integration	Production Readiness Review (PRR)	Test Readiness Review (TRR)	System Verification Review (SVR)	Functional Configuration Audit (FCA)	Physical Configuration Audit (PCA)	Phase D2: Fielding & Checkout	Mission Readiness Review (MRR)	Flight Readiness Review (FRR)	Launch Readiness Review (LRR)	Phase D3: Operations & Disposal	Post Flight Review (PFR)
4	Implement-Software																						
3	Define-Hardware						X																
3	Deploy-Software																						
3	Design-Evaluate-Hardware																						
3	Design-Segment																						
3	Design-System																						
3	Evaluate-Deploy-Hardware																						
3	Evaluate-Hardware-Software																						
2	Define-Data																						
2	Deploy-Subsystem																						
2	Evaluate-Deploy-Requirement																						
2	Evaluate-Hardware-Data																						
2	Evaluate-Software-Requirement																						
2	Evaluate-Subsystem																						
2	Manage-Data																						
1	Define-Data-Requirement																						
1	Define-Design-Implement-Evaluate-Rqmt																						
1	Define-Design-Requirement																						
1	Define-Design-Software																						
1	Define-Evaluate-Hardware																						
1	Define-Hardware-Requirement																						
1	Define-Implement-Data																						
1	Define-Implement-Evaluate-Hardware																						
1	Define-Implement-Software																						
1	Define-Software																						

		Phase 0: Concept Studies	Alternative System Review (ASR)	Phase A: Concept Development	Integrated Baseline Review (IBR)	System Requirements Review (SRR)	System Design Review (SDR)	Phase B: Preliminary Design	Preliminary Design Review (PDR)	Phase C: Complete Design	Critical Design Review (CDR)	Phase D1: Fabrication & Integration	Production Readiness Review (PRR)	Test Readiness Review (TRR)	System Verification Review (SVR)	Functional Configuration Audit (FCA)	Physical Configuration Audit (PCA)	Phase D2: Fielding & Checkout	Mission Readiness Review (MRR)	Flight Readiness Review (FRR)	Launch Readiness Review (LRR)	Phase D3: Operations & Disposal	Post Flight Review (PFR)
1	Deploy-Evaluate-Hardware-Requirement																						
1	Deploy-System-Segment-Requirement																						
1	Design-Data																						
1	Design-Deploy-Segment																						
1	Design-Deploy-Software																						
1	Design-Evaluate-System-Hardware-Rqmt																						
1	Design-Hardware-Data																						
1	Design-Implement-Deploy-Mng-Hardware																						
1	Design-Implement-Hardware																						
1	Design-Segment-Requirement																						
1	Design-Subsystem																						
1	Evaluate-Deploy-Data-Requirement																						
1	Evaluate-Deploy-Hardware-Requirement																						
1	Evaluate-Hardware-Requirement																						
1	Evaluate-Manage-Requirement																						
1	Implement-Evaluate-Hardware-Requirement																						
1	Implement-Evaluate-Software																						
1	Implement-Evaluate-Software-Requirement																						
1	Implement-Hardware-Requirement																						
1	Implement-Segment																						
1	Implement-Software-Requirement																						
1	Implement-Subsystem																						
1	Implement-System																						
1	Manage-Data-Requirement																						
1	Manage-Hardware-Software																						

		Phase 0: Concept Studies	Alternative System Review (ASR)	Phase A: Concept Development	Integrated Baseline Review (IBR)	System Requirements Review (SRR)	System Design Review (SDR)	Phase B: Preliminary Design	Preliminary Design Review (PDR)	Phase C: Complete Design	Critical Design Review (CDR)	Phase D1: Fabrication & Integration	Production Readiness Review (PRR)	Test Readiness Review (TRR)	System Verification Review (SVR)	Functional Configuration Audit (FCA)	Physical Configuration Audit (PCA)	Phase D2: Fielding & Checkout	Mission Readiness Review (MRR)	Flight Readiness Review (FRR)	Launch Readiness Review (LRR)	Phase D3: Operations & Disposal	Post Flight Review (PFR)
1	Manage-Software																						
1	Manage-Subsystem																						
1	Manage-System																						
291	TOTAL		0		0	0	1		0		0		0	0	0	0	0		0	0	0		0
F: People-Product																							
2	Supplier-Requirements																						
1	Acquirer-Data																						
1	Developer-Data																						
1	Developer-Requirements															X	X						
1	Operator-System					X										X	X						
1	Stakeholder-Hardware					X			X														
7	TOTAL		0		0	2	0		1		0		0	0	0	2	2		0	0	0		0
G: People-Process-Product																							
7	Developer-Evaluate-Hardware																						
7	Developer-Implement-Hardware																						
4	Developer-Design-Hardware								X														
4	Developer-Manage-Requirement																						
3	Developer-Define-Requirement				X				X				X										
3	Developer-Deploy-Hardware												X										
3	Developer-Design-Requirement					X																	
3	Developer-Evaluate-Requirement																						
3	Operator-Deploy-Hardware																						
2	Developer-Deploy-Requirement																						
2	Developer-Design-Software								X														

		Phase 0: Concept Studies	Alternative System Review (ASR)	Phase A: Concept Development	Integrated Baseline Review (IBR)	System Requirements Review (SRR)	System Design Review (SDR)	Phase B: Preliminary Design	Preliminary Design Review (PDR)	Phase C: Complete Design	Critical Design Review (CDR)	Phase D1: Fabrication & Integration	Production Readiness Review (PRR)	Test Readiness Review (TRR)	System Verification Review (SVR)	Functional Configuration Audit (FCA)	Physical Configuration Audit (PCA)	Phase D2: Fielding & Checkout	Mission Readiness Review (MRR)	Flight Readiness Review (FRR)	Launch Readiness Review (LRR)	Phase D3: Operations & Disposal	Post Flight Review (PFR)
2	Developer-Evaluate-Subsystem				X																		
2	Developer-Manage-Hardware																						
2	Operator-Deploy-Requirement																						
2	Operator-Evaluate-Requirement																						
2	Supplier-Implement-Hardware																						
1	Acquirer-Evaluate-Requirement																						
1	Acquirer-Evaluate-Subsystem				X																		
1	Acquirer-Implement-Software				X				X				X										
1	Acquirer-Manage-Requirement				X				X				X			X							
1	Developer-Define-Data																						
1	Developer-Define-Evaluate-Requirement				X								X										
1	Developer-Define-Manage-Requirement				X								X										
1	Developer-Deploy-Segment												X										
1	Developer-Deploy-Software				X				X														
1	Developer-Design-Data				X	X																	
1	Developer-Design-Hardware-Software				X	X			X				X										
1	Developer-Evaluate-Data					X																	
1	Developer-Evaluate-Hardware-Requirement																						
1	Developer-Evaluate-Hardware-Software																						
1	Developer-Evaluate-Software																						
1	Developer-Evaluate-System				X																		
1	Developer-Evaluate-System-Hardware																						
1	Developer-Implement-Evaluate-Mng-Software																						
1	Developer-Implement-Requirement																						
1	Developer-Implement-Software																						

		Phase 0: Concept Studies	Alternative System Review (ASR)	Phase A: Concept Development	Integrated Baseline Review (IBR)	System Requirements Review (SRR)	System Design Review (SDR)	Phase B: Preliminary Design	Preliminary Design Review (PDR)	Phase C: Complete Design	Critical Design Review (CDR)	Phase D1: Fabrication & Integration	Production Readiness Review (PRR)	Test Readiness Review (TRR)	System Verification Review (SVR)	Functional Configuration Audit (FCA)	Physical Configuration Audit (PCA)	Phase D2: Fielding & Checkout	Mission Readiness Review (MRR)	Flight Readiness Review (FRR)	Launch Readiness Review (LRR)	Phase D3: Operations & Disposal	Post Flight Review (PFR)
1	Developer-Manage-Data																						
1	Developer-Operator-Deploy-Requirement																						
1	Operator-Define-Requirement																						
1	Operator-Evaluate-Data																						
1	Operator-Manage-Hardware																						
1	Operator-Manage-Requirement																						
1	Stakeholder-Define-Segment																						
1	Stakeholder-Deploy-Hardware																						
1	Stakeholder-Evaluate-Requirement																						
1	Stakeholder-Manage-Requirement																						
1	Supplier-Define-Hardware																						
1	Supplier-Evaluate-Hardware																						
1	Supplier-Evaluate-Requirement																						
1	User-Deploy-Data																						
85	**TOTAL**		0		11	4	0		7		0		8	0	0	1	0		0	0	0		0

Appendix D: Space System Acquisition TR&A Criteria (23:156-166)

Manufacturing Management/Production Capability Review (MM/PCR)
An MM/PCR is conducted during source selection by the government program office at the prospective contractors' facilities to evaluate competing contractors' capability to meet all immediate and future production requirements of proposed systems.

Integrated Baseline Review (IBR)
The IBR provides a mutual (government, contractor program manager) understanding of the inherent technical and programmatic risks in the contractor's plans, the underlying management control systems, and the required resources to reduce risks to an acceptable level. An IBR also examines consistency among technical, schedule, cost, resource and management risks. IBRs are generally conducted within three months after every program key decision point (KDP) and called for by the government program manager as part of his/her risk management approach. Those risks identified during the IBR should be reviewed and mitigation plans incorporated into risk management planning.

System Requirements Review (SRR)
The SRR determines if the contractor's efforts to understand and translate mission requirements into system requirements and operations concept were adequate, and establishes a formal system requirements baseline down to the element level. This includes summarizing significant potential and known program risks and potential risk mitigation strategies, identifying interfaces with and impact to other systems, describing development and operational test approaches, and addressing the producibility of the proposed design concept. The SRR is generally conducted once per program after a significant number of systems functional requirements have been defined and allocated to appropriate CIs and a significant amount of requirements analysis has been completed. This activity is conducted by the contractor and is generally completed within MAG Phase A (concept exploration) or, at the latest, soon after development contract award (MAG Phase C).

System Design Review (SDR)
The SDR evaluates the contractor's approach for optimization, correlation, completeness, and risk mitigation associated with the allocated technical requirements of the identified CIs and the established system design specification baseline. The SDR also includes examinations of the system functional requirements, external interface control requirements, and preliminary system verification plan. A review of the systems engineering process that allocated the technical requirements and the engineering plan for the design and development phase is also conducted. Basic manufacturing considerations and the production-engineering plan will also be reviewed as consideration of design producibility. Careful examination is conducted of all medium- and high-priority risks from assembly level to segment level, and their reflection to the system level along with companion mitigation strategies.

Preliminary Design Review (PDR)
The PDR evaluates the contractor's technical adequacy, progress, and risk resolution for the selected design-to approach for all CIs, and establishes a CI design baseline down to the assembly level. The PDR demonstrates design compatibility with the performance and engineering specialty requirements of the hardware development specifications. Included is an evaluation of technical risks associated with the manufacturing process/methods and the establishment of the compatibility of the physical and functional interfaces among and between CIs (e.g., units, subsystems, or system), facilities, computer software configuration items (CSCIs), and personnel. The PDR processes allow for an engineering assessment of the technical adequacy of top-level design, testing approach, and CONOPS. PDRs are normally conducted once per program for each CI at the assembly level, subsystem, element, and segment building to the system level as appropriate.

Critical Design Review (CDR)
The CDR evaluates the contractor's detailed system design and the detailed build-to design for each CI (e.g., CSCIs, units, subsystems, or system) to determine if each design meets the allocated functional, performance, and engineering specialty requirements. The CDR also is used to evaluate whether the design can be produced and verified, has interface compatibility between CI/CSCIs, facilities, and personnel, and that all risks have been identified, rated, and satisfactory mitigation plans established. CDRs are normally held once per program during
MAG Phase C for each CI (assembly level), subsystem, element, and segment building to a system level, as appropriate.

Test Readiness Review (TRR)
The TRR examines the contractor's progress and status for each CI/CSCI to determine whether hardware and software procedures are complete and the contractor is prepared to start testing. The results of any informal testing and changes to the CONOPS are also reviewed.

Formal Qualification Reviews (FQR)
An FQR evaluates the test, inspection, or analytical results by which a group of hardware configuration items (HWCIs)/CSCIs comprising a system is verified to have met specific performance requirements (specifications or the equivalent). This review does not apply to hardware or software verified at functional configuration audit (FCA) for individual CIs.

Production Readiness Review (PRR)
The PRR evaluates the contractor and the contractor's design readiness to begin manufacturing.23 The PRR is conducted by the government program office and supported by the contractor. The PRR is held incrementally (generally three sessions–two preliminary and one final) during full-scale development. This review is intended to determine if the issues, risks, and corrective actions for manufacturing have been satisfactorily resolved prior to a production go-ahead decision. As the design matures, the review become

more focused and refined dealing with production planning, facilities, allocation, identification and fabrication of tools/test equipment, long lead acquisitions, and the incorporation of producibility-oriented changes.

System Verification Review (SVR)

The SVR incrementally demonstrates that the total system (personnel, products, and processes) is verified to satisfy requirements in the functional and allocated configuration documentation and to confirm readiness for production, support, training, operations, subsequent verifications, additional development, and disposal. The SVR determines if the system produced is capable of meeting the technical performance requirements established in the specifications and test plans.

Hardware Reviews

Acceptance reviews can be informal or formal reviews chaired and presented by the contractor. Formal reviews are sometimes called hardware acceptance reviews or buy-off reviews with the objective of verifying that all hardware, parts, materials, and components have been manufactured and tested in accordance with current design documentation, test procedures, and related documentation prior to government acceptance via a DD-250 and/or delivery to the next highest level assembly or to the launch site. The manufacturing, inspection, and acceptance verifications plus hardware pedigree status are the principal inputs to this review. The team reviews all acceptance test data, and any perceived shortcomings are investigated. The responsible test engineers are available to explain how the test was conducted and anomalies were resolved. Independent pedigree reviews by a government team often supplement contractor-led acceptance reviews and focus on individual critical components and subsystems to establish that the as-built hardware agrees with its design and manufacturing requirements and is not "out-of-family" with predecessors. The pedigree includes a review of manufacturing and quality assurance documentation to verify documented procedures and processes were followed, that any out-of-sequence work maintained the product's integrity, engineering changes were proper, deviations and "use as is" material review board decisions were adequately justified, and whether new processes, materials, or design changes were made that did not violate the product's qualification status. The pedigree also includes an assessment of acceptance testing to ensure procedures were followed, deviations were justified and the root cause of noted test discrepancies was identified with the appropriate corrective action taken.

Functional Configuration Audit (FCA)

The FCA is a formal audit conducted by the government program office and supported by the contractor to demonstrate that hardware and/or software CIs have been achieved. This audit examines the CONOPS, test plans, analysis and inspection reports, as-used qualification test procedures, test data, test reports, drawings, and other supporting documentation. An FCA is conducted on either the first production unit or a pre-production representative of the configuration to be released as an operational production unit. The final FCA occurs at the completion of CI qualification testing.

Physical Configuration Audit (PCA)

The PCA is a formal audit conducted by the government program office and supported by the contractor. The PCA technically examines subject CIs to verify that each CI "as-built" conforms to the technical documentation defining the CI in order to establish the product baseline. A complete PCA is done on the first production unit and is not repeated unless significant engineering changes and resulting modifications to the CI have occurred. Customer formal acceptance of product specification and successful completion of the PCA results in the establishment of the product baseline. The PCA includes a detailed examination of engineering drawings, specifications, technical data, acceptance test procedures and test data, design documentation, and all operational support documentation (e.g., user manuals, diagnostic manuals, and firmware support manuals).

Preliminary Design Audit (PDA)

PDAs are working-level meetings between the government program office team and the contractor prior to the program's formal PDR milestone. PDAs address design thoroughness (ability to meet all functional, performance, and interface requirements from the system to the CI level) in specific functional areas, units, or subsystems, and are milestones on the program's detailed schedule. For complex NSS systems, successful PDAs represent entrance gates to the formal PDR. A series of detailed technical meetings between the contractor, subcontractors, suppliers, and government program office constitutes a single PDA. PDAs are held for each CI (assembly level), subsystem, element, and segment building to the system level, as appropriate. The PDA process allows for very detailed design investigations to ensure requirements can be satisfied, identifies faults/failure modes and plausible mitigation approaches, examines relevant risk mitigation plans and progress, and identifies issues that need to be resolved before the formal PDR. PDAs are normally held once per program prior to the formal PDR in MAG Phase C.

Critical Design Audit (CDA*)*

CDAs are detailed technical working-level meetings between the government program office, the contractor, the subcontractors, and the suppliers prior to the program's formal CDR milestone. For complex NSS systems, CDAs are held for each CI (assembly level), subsystem, element, and segment build to the system level, as appropriate. CDAs address design thoroughness (the ability to meet all functional, performance, and interface requirements from the system to the CI level), risk reduction, and verification and test planning for each level of assembly under examination. During detailed CDA engineering interactions, confidence is gained that the design trades are completed, the final design is complete and producible, and the design has been documented for manufacturing or procurement to begin. Successful completion of each CDA will ensure that all outstanding problems, issues, and risks have appropriate work-off plans. Successful completion of each CDA is an entrance criterion for the program's formal CDR milestone. CDAs are normally held once per program prior to the formal CDR during MAG Phase C.

Readiness Reviews

Readiness reviews provide a formal mechanism that supports the decision-making process by forcing a careful examination of all elements of the system at key maturity milestones relative to final integration, testing, and operator proficiency, including outstanding problems or liens, in preparation for launch. Key decision points (KDPs) include the decision to ship the launch and/or space vehicle to the launch site from the factory; the decision to proceed with vehicle erection on the launch pad; and the decision to proceed with the launch after successfully completing launch integration and processing, successfully demonstrating end-to-end mission connectivity, and successfully demonstrating personnel proficiency through rehearsals. Post-launch reviews are also included to assess flight performance and gather lessons learned.

Independent Readiness Review Team (IRRT)/Mission Assurance Team (MAT)
IRRT/MAT reviews taken together are independent, technical examinations of space vehicle and/or launch vehicle risks beginning approximately one to two years prior to launch. The IRRT is co-led by Space and Missile Systems Center (SMC) and Aerospace, focusing on the integrated launch vehicle. The MAT is co-led by the National Reconnaissance Office (NRO) and Aerospace and focuses solely on the launch vehicle. Both reviews are conducted by a core team, augmented as needed to provide a complete set of discipline and subsystem experts from Aerospace, system engineering and technical assistance (SETA), government, and contractor personnel. The reviews provide technical assessments of the space vehicle or launch vehicle, identify increased risks beyond the established mission baseline to safety or mission success, recommend risk mitigation or confidence-enhancing steps, and evaluate all open issues and the acceptability of all indicated closure paths. The reviews are done incrementally with the final review (IRRT or MAT) occurring before launch. As such, the extent of each review is negotiable depending on the hardware/software design and development stage of the program, hardware/software performance history, and resources available for the review, changes since the last review and the scope of the last review. The timing of final MAT/IRRT review should provide sufficient time for a complete review and for any corrective actions to take place and critical recommendations implemented.

Mission Readiness Review (MRR)
The MRR is a formal review organized by the spacecraft single manager (SM) to evaluate the readiness of the spacecraft before final launch integration activities are initiated. The mission director, launch program SM, and appropriate launch base detachment commander may choose to attend. Program and support organization personnel conduct the MRR, which is supported by the appropriate contractors. Findings and deficiencies should be corrected or disposed of before the flight readiness review (FRR) one to two days before launch. The MRR addresses all system components of mission readiness, including status of flight hardware (spacecraft, launch vehicle, upper stage), launch and support facilities, range and orbital operations, ground station operations, and the readiness and training of all personnel, including customer elements processing mission data. Successful completion of the MRR results in a decision to ship the launch vehicle or space vehicle to the launch base to begin launch processing (i.e., “consent to ship”).

Aerospace President’s Readiness Review
In support of the SMC commander’s FRR (see the following paragraph), the president of The Aerospace Corporation conducts his own objective review of the space and launch vehicles’ readiness to support the designated mission. Both the Aerospace program offices and the IRRT present their findings during this review and support more detailed technical discussions on specific issues, as required by, prior to, during, or subsequent to the president’s formal review. Aerospace corporate vice presidents of the appropriate Space Launch Operations, Space Program Operations or National Systems Group, or Engineering Technology Group support the president’s review. In accordance with SMCI 63-1201, the president presents his findings to the SMC commander during the FRR and participates in the readiness poll.

Pre-ship Review (PSR)
The program conducts hardware PSR to assure that flight hardware and components, software, ground support equipment, and procedural documentation are ready to ship to the deployment site. Operations personnel participate in this review. This type of review is meant to identify any open issues affecting deployment and subsequent operations, verify that planning is in place to close-out these issues in a timely manner, and verify supportability of the program’s ensuing activities. Operations personnel ensure sufficient coordination between the system contractor and Range/launch site (and/or any other receiving site), to assure that the latter is ready to receive program hardware, receiving support has been appropriately scheduled, and receiving facilities are prepared to support hardware arrival and post-shipping inspection activities.

Flight Readiness Review (FRR)
The FRR is a formal review organized and coordinated with applicable government program offices and presented to the SMC commander (or designated representative) by the mission director and supported by the launch base and appropriate contractors. The FRR evaluates the space flight worthiness of the integrated flight hardware (space vehicle, upper stage and launch vehicle) approximately one to three weeks before launch. It also addresses the readiness of launch and support facilities (ground systems), range and orbital operations, and the readiness and training of the operating personnel. The review includes a safety verification of the integrated system.
The objective is to ensure the prime contractors, The Aerospace Corporation, the spacecraft program office, launch programs, and the SMC commander agree that the launch vehicle is flight worthy and ready to begin final launch operations. Other inputs to the FRR include the IRRT and MAT reviews, the contractor and Aerospace presidents’ reviews, and detailed briefings by both the spacecraft and launch program teams. At completion of the FRR, the SMC commander will assess and may certify space flight worthiness of the integrated system for USAF space missions. For USAF-managed space and launch vehicles in support of non-USAF customers, the SMC commander will be responsible for approving the SM’s certification. For selected critical missions, the SMC commander will follow-up with an executive mission readiness report (EMRR) to Air Force senior leadership. The FRR is conducted after the launch vehicle and spacecraft are integrated, approximately one to two weeks before launch.

Launch Readiness Review (LRR)
A LRR is an operations readiness review organized by the Launch Decision Authority (i.e., launch base wing commander, or the Launch Processing Agency when a non-Air Force Space Command launch site is used) and supported by the appropriate contractors. It is conducted following the integrated launch and space vehicle systems test one or two days before launch. The LRR process

provides a summary prelaunch assessment of the readiness status of the total system (space and launch vehicle), the launch facility, range safety and instrumentation, the Air Force Satellite Control Network, the operational mission control station, operations personnel, and other launch or on-orbit support. Launch Decision Authority also verifies the closure of issues and items and determines the readiness status of safety, training, weather, and recovery teams.

Post-flight Review (PFR)

A PFR is conducted for all missions requiring a MRR and the results are presented to the single manager who chaired the MRR. It is intended as a top-level summary predicated on post-launch, in-depth assessments conducted by the space vehicle program manager, launch vehicle program manager, and appropriate payload mission managers. The PFR typically covers the time from the MRR through early on-orbit operations. The PFR addresses pre-launch ground operations, launch operations, mission and space vehicle operations, the launch vehicle, the space vehicle, critical ground systems and interfaces, and the payload user's ground interface to receive and process mission data. The PFR captures all Lessons Learned from the mission and provides both feedback and schedule imperative to the government program office to implement Lessons Learned before the program office's next mission. PRRs are held approximately 60 days after launch and early on-orbit testing is completed.

Bibliography

1. Abalateo, Mila D. and Lee, Joni R. *Benchmarking Practices of Air Cargo Carriers: A Case Study Approach*. Air Force Institute of Technology (AFIT): Center of Systems Engineering, AFIT/GLM/LAL/93S-1, 1993.

2. *Adoption of ISO/IEC 15288:2002 Systems Engineering – System Life Cycle Process*. Institute of Electrical and Electronic Engineers (IEEE) 15288 Standard, 2004.

3. *Application and Management of the Systems Engineering Process*. Institute of Electrical and Electronic Engineers (IEEE) 1220 Standard, 2005.

4. Balasubramanian, Krishnakumar; Schmidt, Douglas C.; Molnar, Zoltan and Ledeczi, Akos. *System Integration Using Model-Driven Engineering*. Institute for Software Integrated Systems, Vanderbilt University, Nashville: <email: {kitty, schmidt, zolmol, akos}@isis.vanderbilt.edu>

5. Blanchard, Benjamin S. and Fabrychy, Walter J. *Systems Engineering and Analysis*, *3rd Edition*. Upper Saddle River, NJ: Prentice Hall, Inc., 1981.

6. Blanchard, Benjamin S. *System Engineering Management, 3rd Edition*. Hoboken, NJ: John Wiley & Sons, Inc., 2004.

7. Boehm, Barry and Lane, Jo Ann. *Better Management of Development Risks: Early Feasibility Evidence*. 7th Annual Conference on Systems Engineering Research (CSER), 2009. University of Southern California: <email: boehm@usc.edu, jolane@usc.edu>

8. Boehm, Barry and Lane, Jo Ann. *Using the Incremental Commitment Model to Integrate System Acquisition, Systems Engineering, and Software Engineering*. University of Southern California: <email: boehm@usc.edu, jolane@usc.edu>

9. Buede, Dennis M. *The Engineering Design of Systems – Models and Methods*. New York, NY: John Wiley & Sons, Inc., 2000.

10. Busch, Carol; De Maret, Paul S.; Flynn, Teresa; Kellum, Rachel; Le, Sheri; Meyers, Brad; Saunders, Matt; White, Robert and Palmquist, Mike. *Content Analysis*. Colorado State University Department of English, Writing@CSU, 2005. Retrieved from http://writing.colostate.edu/guides/research/content/.

11. *Capability Maturity Model® Integration (CMMI℠), Version 1.1*. Carnegie Mellon Software Engineering Institute, 2002.

12. Cheng, Paul G. and Smith, Patrick. *Learning from Other People's Mistakes.* Crosslink: The Aerospace Corporation Magazine, Fall 2007.

13. Cheng, Paul G. TOR-2005(8617)-4204, *100 Questions for Technical Review*. The Aerospace Corporation, 2005.

14. Collens, Josiah R. Jr. and Krause, Bob. *Theater Battle Management Core System (TBMCS), Systems Engineering Case Study*. Air Force Institute of Technology (AFIT): Center of Systems Engineering, February 2005.

15. *Committee on Pre-Milestone A Systems Engineering: A Retrospective Review and Benefits for Future Air Force Systems Acquisition*. National Research Council (NRC).

16. *Defense Acquisition System*. Department of Defense Directive (DoDD) 5000.01, 2007.

17. *Defense Acquisition System*. Department of Defense Directive (DoDD) 5000.1, 2003. (Superseded).

18. *DOD's Goals for Resolving Space Based Infrared System Software Problems Are Ambitious*, GAO Report to Congressional Committees, United States Government Accountability Office. GAO-08-1073, September 2008.

19. Eisner, Howard. *Essentials of Project and Systems Engineering Management, 2nd Edition*. New York, NY: John Wiley & Sons, Inc., 2002.

20. *Global Positioning System Systems Engineering Case Study*. Air Force Institute of Technology (AFIT): Center of Systems Engineering, October 2007.

21. Grady, Jeffrey O. *System Integration*. Boca Raton, FL: CRC Press LLC, 1994.

22. Grady, Jeffrey O. *System Requirements Analysis*. New York, NY: McGraw Hill, Inc., 1993.

23. Guarro, S.B. and Tosney, W.F. (Eds). TOR-2007(8546)-6018, *Mission Assurance Guide*. The Aerospace Corporation, 2005.

24. *Guidance for DoD Space System Acquisition Process*. National Security Space-Systems (NSS) Acquisition Policy NSSAP 03-01, 2004.

25. Haskins, Cecilia; Forsberg, Kevin and Krueger, Michael. *Systems Engineering Handbook - A Guide for System Life Cycle Processes and Activities*. International Council on Systems Engineering (INCOSE), 2007.

26. *Integrated Product and Process Development (IPPD) Handbook*. Department of Defense, 1998.

27. *Interim Defense Acquisition Guidebook (DAG)*. Department of Defense, 2009.

28. Kossiakoff, Alexander and Sweet, William N. *Systems Engineering Principles and Practice*. Hoboken, NJ: John Wiley & Sons, Inc., 2003.

29. Maier, Mark W. and Rechtin, Eberhardt. *The Art of Systems Architectin,. 2nd Edition*. Boca Raton, FL: CRC Press LLC, 2002.

30. Martin, Richard A. *International Standards for System Integration*. International Council on Systems Engineering (INCOSE), 2005. Tinwisle Corporation: <e-mail: tinwisle@bloomington.in.us>

31. Mattice, James. *Hubble Space Telescope Systems Engineering Case Study*. Air Force Institute of Technology (AFIT): Center of Systems Engineering, 2005.

32. *Military Standard for Systems Engineering, Draft*. Department of Defense, HQ AFMC/ENS, MIL-STD-499B, 1994. (This standard was not signed by the Department of Defense).

33. Muller, Gerrit. *Coping with System Integration Challenges in Large Complex Environments*. International Council on Systems Engineering (INCOSE) 17th Annual International Symposium Proceedings, 2007. Embedded Systems Institute: <e-mail: gerrit.muller@embeddedsystems.nl>

34. Myerson, Judith M. (Ed). *Enterprise Systems Integration, Second Edition*. Boca Raton, FL: Auerbach Publications, 2002.

35. *NAS System Engineering Manual*, *Version 3.1*. Institute for Telecommunications, U.S. Department of Commerce, June 2006.

36. *National Consensus Standard for Configuration Management*. Electronic Industry Alliance (EIA) 649, 1998.

37. *Operation of the Defense Acquisition System*. Department of Defense Instruction (DoDI) 5000.02, 2008.

38. *Operation of the Defense Acquisition System*. Department of Defense Instruction (DoDI) 5000.2, 2003. (Superseded).

39. *Operational Suitability, Safety, & Effectiveness Process*. Space & Missiles Systems Center (SMC) Instruction (SMCI) 63-1201, U.S. Air Force.

40. Pennel, L.W. and Knight, F.L (Eds). TOR-2005(8583)-3, *Systems Engineering*. The Aerospace Corporation, 2005.

41. Roche, James G., Secretary of the Air Force. Air Force Times: June 24, 2002.

42. Sage, Andrew P. and Rouse, William B. (Eds), and Palmer, James D. (Author). "Systems Integration." *Handbook of Systems Engineering and Management*. New York, NY: John Wiley & Sons, Inc., 1999.

43. Sage, Andrew P. *Systems Engineering*. New York, NY: John Wiley & Sons, Inc., 1992.

44. Sambur, Marvin. *Incentivizing Contractors for Better Systems Engineering*. Secretary of the Air Force for Acquisitions (SAF/AQ) Memo, April 2003.

45. Sellers, Jerry Jon. *Understanding Space – An Introduction to Astronautics, 3rd Edition*. McGraw Hill, Inc., 2005.

46. Shishko, R. and Chamberlain, R.G. *Systems Engineering Handbook*. National Aeronautics and Space Administration (NASA) SP-610S, 1995.

47. Simpson, Joseph J. and Simpson, Mary J. *System Integration Frameworks*. International Council on Systems Engineering (INCOSE), 2005. System Concepts: <e-mail: jjs-sbw@eskimo.com>

48. Singletary, Lester A. *Empirical Study of Attributes and Perceived Benefits of Applications Integration for Enterprise Systems*. Louisiana State University, August 2003.

49. *Space-Based Infrared System-low at Risk of Missing Initial Deployment Date*, Report to the Chairman, Subcommittee on Defense, Committee on Appropriations, House of Representatives, United States Government Accountability Office, GAO-01-6, February 2001.

50. Stockman, Bill and Fornell, Gordon E. *Peacekeeper Intercontinental Ballistic Missile System Engineering Case Study*. Air Force Institute of Technology (AFIT): Center of Systems Engineering.

51. Stutzke, Richard D. *Factors Affecting Effort to Integrate and Test a System of Systems*. 20th International COCOMO and Software Cost Modeling Forum, 2005. Science Applications International Corporation: <e-mail: Richard.d.stutzke@saic.com>

52. *Systems Engineering Case Studies Synopsis of the Learning Principles*. Air Force Institute of Technology (AFIT): Center of Systems Engineering, 2008.

53. *Systems Engineering Fundamentals*. Defense Acquisition University (DAU), 2001.

54. *Systems Engineering Guide, Version 1.2*. Aeronautical Systems Center (ASC), Department of the Air Force, 2004.

55. *Systems Engineering Primer & Handbook - Concepts, Processes, and Techniques, 3rd Edition*. Space & Missiles Systems Center (SMC) U.S. Air Force, 2005. <http://deskbook.dau.mil/jsp/default.jsp>.

56. *Systems Engineering Technical Review Timing*. Naval Air Center. Department of Navy, 2007. <http://www.navair.navy.mil/kms/41g/>.

57. Tahan, Meir and Ben-Asher, Joseph Z. *Multistage System Integration*. International Council on Systems Engineering (INCOSE). Faculty of Aerospace Engineering.

58. Thomas, L. Dale, Cohen, Brad, and Young, Judy. *System Integration Applications of Information Systems in the Space Station Freedom Program*.

59. Valerdi, Ricardo; Boehm, Barry W. and Reifer, Donald J. *COSYSMO: A Constructive Systems Engineering Cost Model Coming of Age*. International Council on Systems Engineering (INCOSE) 13th Annual International Symposium Proceedings, 2003. University of Southern California: <e-mail: rvalerdi@sunset.usc.edu, boehm@sunset.usc.edu, dreifer@earthlink.net>

60. Valerdi, Ricardo; Rieff, John E.; Roedler, Garry J.; Wheaton, Marilee J. and Wang, Gan. *Lessons Learned from Industrial Validation of COSYSMO*. International Council on Systems Engineering (INCOSE) 17th Annual International Symposium Proceedings, 2007. <e-mail: rvalerdi@mit.edu, john_e_rieff@raytheon.com, garry.j.roedler@lmco.com, marilee.wheaton@aero.org, gan.wang@baesystems.com>

61. Wasson, Charles S. *System Analysis, Design, and Development Concepts, Principles, and Practices*. Hoboken, NJ: John Wiley & Sons, Inc., 2006.

62. Wertz, James R., and Larson, Wiley J. *Space Mission Analysis and Design, 3rd Edition*. Hawthorne, CA: Microcosm Press, Inc., 2007.

Vita

Douglas R. Dillon has more than 29 years experience in Electrical Engineering. He has experience working with communications equipment, test equipment, and a variety of DoD weapon systems. As a Marine Corps officer, Mr. Dillon was trained as a Communications Officer and engineered tactical communications architectures. After receiving a Masters of Science in Electrical Engineering with a specialty in Communications Engineering, he was assigned as a Program Manager for Marine Corps transportable communications systems. At Quantico, Virginia, Captain. Dillon performed extensive engineering on a 1kWatt High Frequency transportable communications system, the first to use Automatic Link Establishment (ALE) technology. After retiring from the Marine Corps, Mr. Dillon moved back to Colorado where he took a job working for the National Test Facility (NTF) at Falcon AFB (now Schriever AFB) where he was tasked with National Missile Defense simulation software development. In 1996, Mr. Dillon took a job with the 50th Space Wing, Schriever AFB assigned as a Test and Evaluation Engineer. During his tour with the 50th, he performed OT&E on a variety of Air Force systems, most notably, Air Force Satellite Control Network (AFSCN) tracking stations and Satellite Collision Avoidance technologies. After transfer to the 17th Test Squadron at Schriever AFB, Mr. Dillon supported numerous operational tests on Air Force space systems until he moved to the United States Space Command (now disbanded and mission sent to USSTRATCOM). While at US Space Command, he was assigned as a deliberate planner tasked with drafting a computer network defense contingency plan (CONPLAN 3900). In 2001, US Space Command was directed to disband and transfer their mission and resources to

USSTRATCOM. At this time, Mr. Dillon took a job with Space and Missile Systems Center (SMC) Detachment 11, working as an Electronics Engineer. Although the organizational names have changed over time, Mr. Dillon continues to work for SMC as a Lead Engineer for the Weather Sustainment Division.

Enriqueta Mariela Styers has more than 25 years experience in the Department of Defense (DoD). Right after college, she started with the Department of Navy in the area of Automatic Test Equipment (ATE) maintaining a software tool that supports fault isolation of digital circuit cards. After 5 years, she continues her career with the Department of Air Force to employ the software tool on aircraft circuit cards that need repairing. Although in the same area, Ms. Styers' experience extended to developing hardware that interfaces the digital circuit card to the ATE that runs the software tool. The next 10 years, her experience revolved around this area of the aircraft world from developing test program sets, training and mentoring users of the software tool, maintenance of the software tool unto computer network administration and customer service. Having enough of that, Ms. Styers moves on to space system programs in the area of satellite communications reverting back to software support of a communication terminal. In search of excellence, she was able to move into systems engineering supporting the different systems in different satellite communication programs. As part of this enrichment, she pursues a Masters of Science in Systems Engineering (Space Track) with the Air Force Institute of Technology (AFIT). This step has enabled Ms. Styers to explore what waits in the "final frontier".

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to an penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
00-12-2009	Master's Thesis	October 2008 – December 2009

4. TITLE AND SUBTITLE

Analyzing Systems Integration Best Practices and Assessment in DoD Space Systems Acquisition

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Dillon, Douglas R., YD-02, AFSPC/SMC/SLG
Styers, Enriqueta Mariela, YD-02, AFSPC/SMC/MC/OSL

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way, Building 640
WPAFB OH 45433-8865

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GSE/ENV/09-02DL

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Louis M. Johnson, DAF – AFSPC/SMC/SLG/CL
James C. Breidenbach, DAF – AFSPC/SMC/MCSW/OSL
1050 E. Stewart Avenue, Bldg. 2025
Peterson AFB, CO 80914-2902

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Senior leadership of the Air Force's Space and Missile Center suggested an investigation of systems integration within the space acquisition community in the fall of 2008. This thesis performs that investigation. A review concluded that while Systems Integration (SI) is extensively discussed as an area deserving considerable attention in the Systems Engineering literature, definitions are weak and methods and tools non-existent. Known SI activities are not being traced and assessed for adequacy throughout system development. Employing the Space System Acquisition Lifecycle Framework as the environment for this research, a method of characterizing and tracing SI throughout a program's lifecycle by using technical reviews and audits (TR&A) is proposed. Subsequent to a SI trace of an acquisition program, an assessment can be performed to determine the adequacy of the integration of Systems Engineering (SE) tasks. Using this assessment, prudent adjustments to program resources (e.g., SE, finance, research and development, program management, etc.) can be considered that will mitigate or resolve program deficiencies caused by insufficient SI. The proposed method is demonstrated across technical reviews and audits of the Global Positioning Systems (GPS) program. The results of this thesis should accentuate the value of SI during space system acquisition - a key consideration which is rarely recognized.

15. SUBJECT TERMS

Systems Integration, Systems Engineering, Space Systems Acquisition, Lessons-Learned, Traceability Matrix

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
					John M. Colombi, Ph.D., AFIT/ENV
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(Include area code)*
U	**U**	**U**	**UU**	158	(937) 255-3355, ext 3347 (john.colombi@afit.edu)

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18